MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

Memo No. 341 December 1975

SPATIAL DISPOSITION OF AXES IN A GENERALIZED CYLINDER REPRESENTATION OF OBJECTS THAT DO NOT ENCOMPASS THE VIEWER

by

D. Marr and H. K. Nishihara

ABSTRACT. It is proposed that the 3-D representation of an object is based primarily on a stick-figure configuration, where each stick represents one or more axes in the object's generalized cylinder representation. The loosely hierarchical description of a stick figure is interpreted by a special-purpose processor, able to maintain two vectors and the gravitational vertical relative to a Cartesian space-frame. It delivers information about the appearance of these vectors, which helps the system to rotate its model into the correct 3-D orientation relative to the viewer during recognition.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract number N00014-75-C-0643.

Summary

1. It is observed that the generalized cylinder representation of a 3-D object generates two distinct problems: describing the cross-sections associated with each axis, and representing the relative dispositions of the axes in space.

2. The second problem amounts to representing the spatial arrangement of a stick figure. A method for doing this is given.

3. The stick-figure is described by a loosely hierarchical assertional database, called a *3-D model*. Use of this database is flexible, and it can support levels of description that cover the spectrum from a very coarse overall summary to very fine detail of one small part.

4. In order to be used, a 3-D model has to be interpreted through an (essentially) analogue mechanism, called the image-space processor. In its minimal implementation, this processor maintains a representation of two directions (called *$axis* and *$spasar*) in a Cartesian space-frame, in additional to the gravitational vertical.

5. The image-space processor's instruction set is small. Its important functions are:

(a) setting the $axis to one of the space-frame's 3 axes or to the gravitational vertical;

(b) setting the $spasar to an arbitrary orientation relative to the $axis, this includes the ability to rotate the $spasar about the $axis;

(c) setting the $axis to the orientation of the $spasar; and

(d) rotating the space-frame about four distinguished axes, its three coordinate-axes and the gravitational vertical. (In a minimal implementation of the image-space processor, the position of the $spasar would have to be reconstructed after a frame rotation, rather than being rotated with it.)

6. The image-space processor can deliver information about the lengths and orientations of the projections of the $axis and $spasar onto the image plane. These help the system to "rotate" its model into the correct 3-D orientation relative to the viewer. Some evidence is given that this can be carried out by a process of relaxation.

7. Fahlman's symbol-mapping problem is dealt with by dividing it into its component problems, and using special techniques for each component. The problem of indexing for recognition is discussed.

8. It is observed that this theory may help to explain various aspects of the psychology of human vision. These include the "mental rotation" experiments of Shepard and his collaborators, and the clinical disabilities described by Warrington & Taylor (1973) that follow right parietal lesions.

Introduction

The two current ideas for representing three-dimensional structures are the "generalized cylinder" representation proposed by T. O. Binford and implemented by Agin (1973), Nevatia (1974), and by Hollerbach (1975); and the "multiple view" representation (Minsky 1975). The generalized cylinder representation of a structure is obtained by specifying its axis and the cross-section at each point along it. Agin and Nevatia used a laser range-finding technique to obtain the generalized cylinder representation of such objects as a barbie doll, a snake, and a horse. Hollerbach studied the representation of a wide range of pottery. The multiple view representation is based on the insight that if one chooses ones primitives correctly (e.g. the "side" of a cube), the number of qualitatively different views of an object may be quite small. A number of important questions of detail remain unanswered because this idea has not yet been implemented, and it remains to be seen whether a theory can be built upon it.

The generalized cylinder represention introduces two main problems: obtaining the axis and a description of the cross-section of the different parts of an object (arms, legs, torso), and representing the spatial disposition of the components thus obtained. The second of these problems has hitherto received no attention, and it is the one that we address here. To solve it, one has to tackle directly the problem of representing the positions of items in three-space, and this article presents a method for doing it which we believe may be of interest to experimental psychologists.

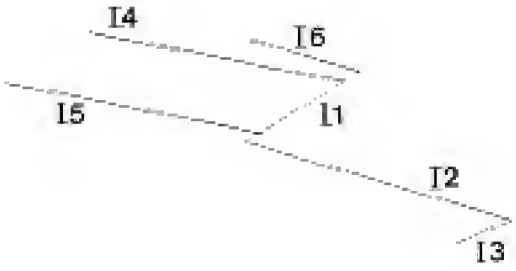
Statement of the problem

The principle of modular design is central to the vision system of which this article describes a part. For example, the processes that define a place-token in an image are almost independent of the processes that subsequently group them (Marr 1975); the processes that select items to be tested for symmetry are similarly independent of the routines that detect it (Marr 1976); the extraction of a form from the primal sketch is often independent of the processes that describe that form (Marr 1975); and much of the segmentation of a form into its generalized cylinder description can apparently proceed (to a first approximation) independently of knowledge about what that form is (Marr & Vaina, to appear). The representation of the three-dimensional structure of an object using generalized cylinders can also be split into the two problems mentioned above, and our first proposition is that the two problems are dealt with by separate modules. One module computes the description of the shape of each component, and another describes the relative spatial dispositions of those components.

From this proposition, it follows that describing the spatial disposition of parts of an object or animal may be reduced to the problem of describing the dispositions of the *axes* that occur in its generalized cylinder description. Thus for animals, our problem reduces to that of describing stick figures - models made out of pipe-cleaners, one for each axis (see figure 1). The vision system being constructed at our laboratory is already capable of computing this description from a raw image in simple cases.

The problem then is to represent the three-dimensional configuration of a

FIGURE 1. The theory asserts that the 3-D representation of a shape is decomposed into two parts, the description of the cross-sections that occur in the shape's generalized cylinder representation, and the disposition of the axes of these cylinders in space. The theory deals with the second problem, which is essentially the problem of describing stick figures. The shapes in these pictures were made out of pipe-cleaners. The reader will have no trouble in recognizing the giraffe, deer, rabbit and ostrich. That their recognition is so easy makes it reasonable to suppose that at some stage, we ourselves decompose the 3-D representation problem into similar components.




```
$0000
SHAPE: $CYLINDER
END2: (169.642866 100.495794)
END1: (44.642862 26.44626351)
IMAGE-NAME: I1
IMAGEL: TRUE

$0001
SHAPE: $CYLINDER
END2: (331.90955 -55.996113)
END1: (46.642862 26.44626351)
IMAGE-NAME: I2
IMAGEL: TRUE

$0002
SHAPE: $BLOB
END2: (271.64821 -80.558851)
END1: (331.90955 -55.996113)
IMAGE-NAME: I3
IMAGEL: TRUE

$0003
SHAPE: NIL
END2: (-118.847223 143.3579443)
END1: (151.785722 89.91729)
IMAGE-NAME: I4
IMAGEL: TRUE

$0004
SHAPE: NIL
END2: (-208.13294 98.4654211)
END1: (62.500005 37.0247683)
IMAGE-NAME: I5
IMAGEL: TRUE

$0005
SHAPE: $STICK
END2: (49.948434 134.8467821)
END1: (169.642866 100.495794)
IMAGE-NAME: I6
IMAGEL: TRUE
```

FIGURE 2. This is the raw data provided to our system from the intermediate visual processor. It consists of a collection of imagels which are descriptions of individual generalized cylinders found in the image. Each imagel has two end points in the image plane and optionally a shape property such as $stick which supplies additional information about the imagel such as average thickness, roundedness, flatness, and so on.

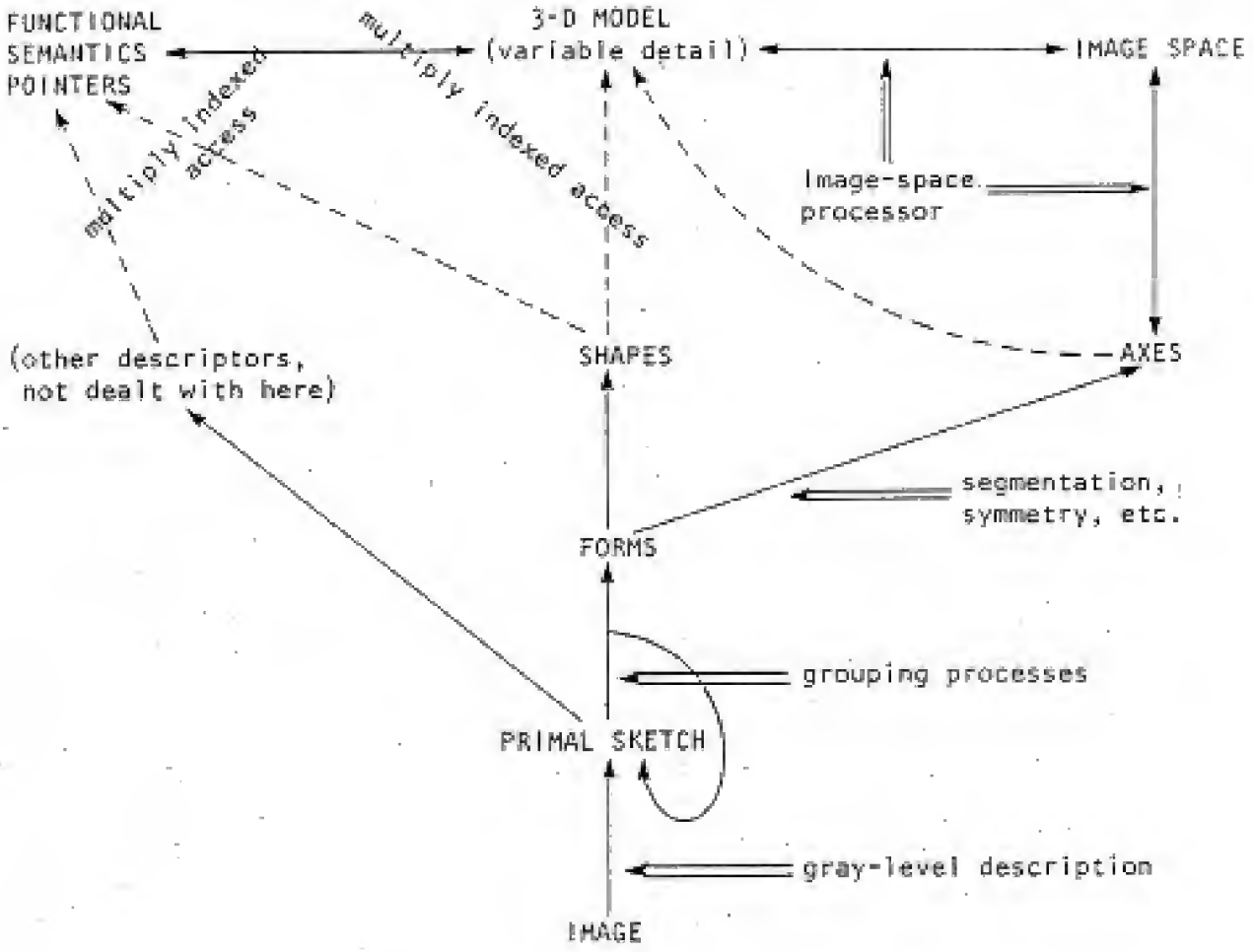


FIGURE 3. This diagram summarizes our overall view of the recognition problem. The important points for the present article are (a) that the representation of 3-D models is quite separate from the representation of functional semantics; (b) indexes exist that give rich access to functional semantics pointers from descriptions at every stage after the separation of figure from ground; and (c) that for difficult images, considerable interaction may have to take place between the description of a form, the image-space processor, and the 3-D model indexer before an appropriate 3-D model is found. When one is found, still another step (of relaxation) may be necessary before the functional semantics indexer acquires enough information to recover the correct pointer.

stored stick figure so as to relate the angles and apparent lengths found in an image like figure 2 to the three-dimensional structure of the object and the perspective from which it is being viewed. We want a solution that in some sense minimizes the computational complexity, but not necessarily the computational power, of the machinery required to implement it.

Background: recognition is not an all-or-none process

Before giving an outline of the theory, we need to make two general points that have deeply influenced the way we approached the recognition problem. The first is that the stored three-dimensional representation of an object is separate from the representation of its functional semantics. This article deals only with the three-dimensional problem; that a horse trots, gallops, eats grass, can be ridden, and is liable to kick are not represented here. It is very reasonable to keep the two separate, because a living horse differs in a fundamental way from a statue of a horse, despite its similar geometry. Nevertheless, there are grounds for thinking that the top-level token that organizes the functional semantics of a horse is the one that is closest to the linguistic label "horse", and part of what we mean by recognition is the ability to address this pointer on viewing an image.

Which brings us to our second point. It is often the case that the functional semantics pointer can be acquired quite early in the analysis of an image - many simple and definite cues exist that can be extracted before a 3-D description has been built. Any indexing strategy for fast recovery of the functional semantics pointer would certainly take advantage of this, which means being sensitive to descriptions at every stage after figure-ground separation. The variety of cues that are available in most images probably means that only rarely will one have to proceed all the way to a 3-D description before a match in the database is found. Indeed, we would expect this to happen only when the object is being seen from a deceiving perspective, or when the prevailing illumination is unusual. It is quite easy to design flexible indexing techniques, that can make use of clues of diverse kinds from different levels.

Our overall picture of the recognition problem is illustrated in figure 3. This makes clear our belief that there are many paths to the functional semantics pointer, some of them fast but not necessarily available from every image, and others that are slower, but which usually guarantee results. In a penetrating analysis, Warrington & Taylor (1973) concluded that the 3-dimensional description and the functional semantics of an item are represented in distinct cortical areas. Their evidence for this assertion is double dissociation between the two kinds of deficit, observed in patients with left (for semantics disorders) and right (for disorders of three-dimensional representation) parietal lesions.

This article is concerned with only one small part of figure 3, namely the construction of a 3-D model of the disposition of the axes in space. In order to be convinced that this is certainly one of the possible paths to the functional semantics pointer, one has only to look at figure 1. Pipe-cleaner animals exhibit *only* the lengths and dispositions of their axes, yet we have no trouble recognizing the giraffe, rabbit or ostrich in this figure.

Outline of the theory

There are four main components to the theory. We give a brief description of them first, so that the reader has an overall framework within which to fit the details.

At some stage in the representation of three-dimensional space, one needs a primitive ability to represent a vector (i.e. a direction and a length). Accordingly, the first component of the theory is a processor that provides this primitive ability. It is called the *image-space processor*, and it can maintain two connected vectors within a supporting space-frame. These two vectors are called the *$axis* and the *$spasar* (an abbreviation for space-arrow). The processor can translate the end of the $spasar to an assigned position on the $axis, can rotate it around the $axis, and can rotate it in the plane containing both vectors. In this way, the $spasar can be brought to an arbitrary relation with the $axis.

In addition to these facilities, the image-space processor can move the $axis to wherever the current $spasar happens to be, and it can act as though a small number of space-frame rotations could be performed. We do not regard the space-frame rotations as true extra facilities, because there are ways of simulating them using only the $axis and $spasar.

The usefulness of the processor for recognition arises from the fact that as well as maintaining the three-dimensional relation between the $axis and the $spasar, it can compute the lengths, directions, and angle between their projections. The computational load attached to doing this is small.

The second component of the theory is a propositional database that represents by assertions useful three-dimensional relations between the axes of the objects being viewed. The datastructure for a single physical object is called a *3-D model*, and its purpose is to explain every image element delivered by earlier visual processes, up to a level of detail appropriate to the circumstances. There are two important points about this database. Firstly, its organization is loosely hierarchical. It can provide descriptions of parts of an object that cover richly the spectrum from a coarse, one-axis description of a whole object, to a fine specification of one small part of it. For example, at the top level, a horse may be representable as a single, horizontal axis. At a lower level, the two forelegs are treated as a single axis. At the next stage, this description decomposes to the left-foreleg and the right-foreleg; and further down, the single axis description of the left-foreleg decomposes to two, splitting at what the layman would call the knee. Figure 4 shows some of the ways in which a typical animal datastructure could be decomposed. The second point is that, three-dimensional positions are represented by local relations between adjacent parts of a body, not by absolute coordinates in a circumscribing frame of reference. Thus the position of a toe is stored relative to a foot, which is stored relative to a leg, which in turn is stored relative to the torso. In order to discover the relationship between the head and the toe, these intermediate relations have to be examined.

The third component of the system is the *interpreter*, whose job is to create and maintain the interface between the database, the image-space processor, and the information being delivered from the image. The interpreter is capable of reading the assertions in a 3-D model, binding the $axis and $spasar in the image-space processor to

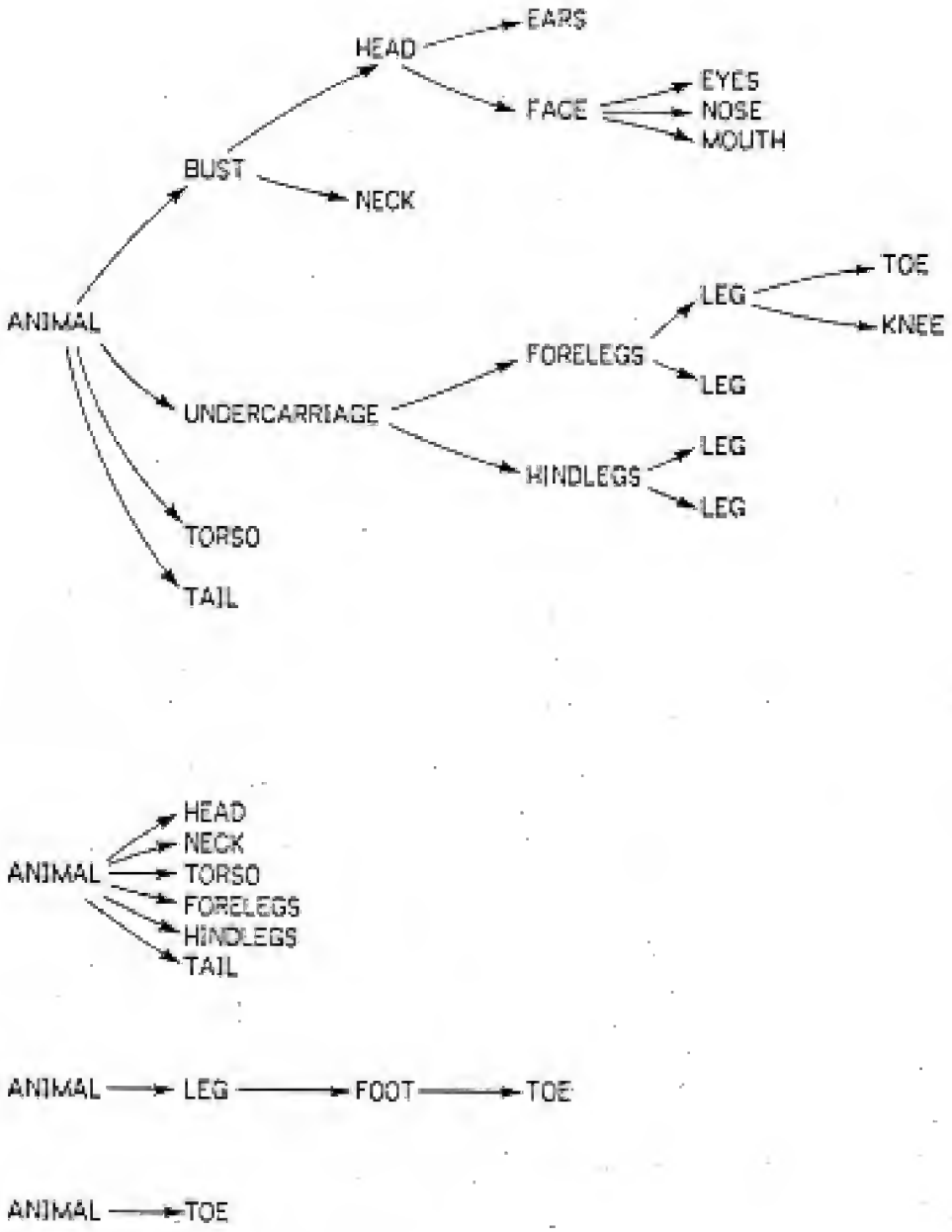


FIGURE 4. The representation of a stick figure is loosely hierarchical, and allows descriptions to be created at many levels of detail. At the top level, a horse may be represented as a single, horizontal axis (to answer questions like "Where is the horse pointing?"). To answer the question "Where is its front left hoof pointing?", the left foreleg will have been unpacked to a considerable degree of detail, while the hindlegs may still be bound to nothing finer than a single HINDLEGS axis. This figure shows some of the ways in which a typical ANIMAL datastructure may be decomposed.

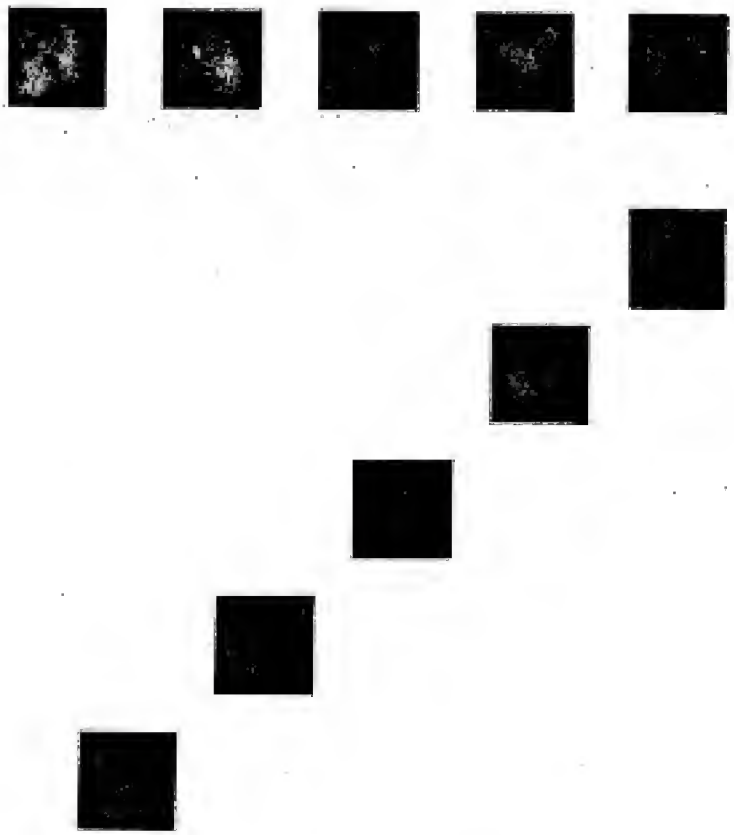
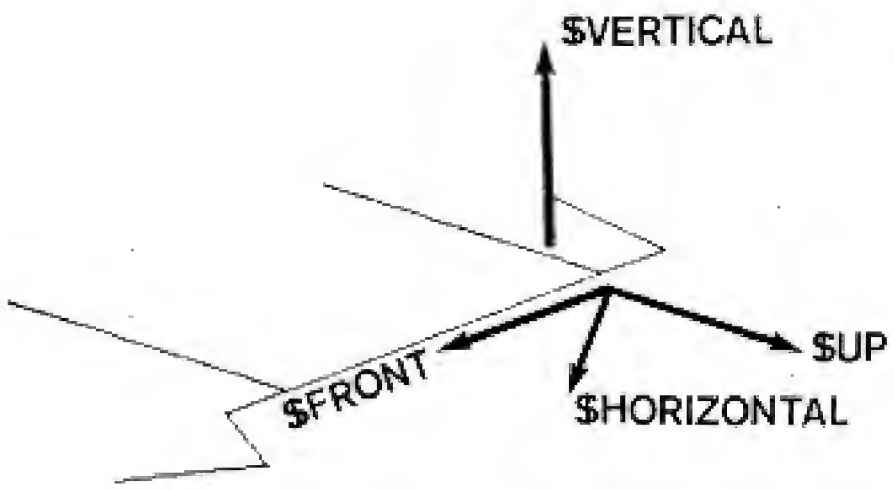


FIGURE 5. This figure, taken from Minsky & Papert (1972), illustrates the influence of an axis on the description of a figure. In one row, the shapes are seen as squares, and in the other, as diamonds. The establishing of axes in a 2-dimensional figure is important for our theory, since it determines how the description of a 2-D configuration is constructed. This figure is the 2-D analog of figure 1, since it establishes that one precondition for using our theory as a psychological model – namely the computation of axes during the analysis of 2-D patterns – is satisfied by our visual systems.



SPACE-FRAME

FIGURE 6. The principle directions defined in the text are displayed graphically in the figure. The image-space processor is capable of simulating rotations about $up, $front, $horizontal and $vertical. After such a rotation, the position of the $spasar (and possibly also of the $axis) must be reconstructed from adjunct relations in the 3-D model.

appropriate axes in the 3-D model, and causing these vectors to be rotated until they have the same 3-D relation to one another as is specified in the model. The interpreter can compare the resulting vector with an image element, and can report on any discrepancies between predicted and measured properties of the image. Various global variables are set and read by the interpreter; they include a certain degree of translational freedom; and an overall scale factor, called the *$scale*, which governs the relation between the size property of an item in the database and the length of the $spasar to which it gives rise. Decisions about which 3-D model to instantiate and what parts of it to concentrate on form the controlling information for the interpreter, which then tries to match the model to the image by three-dimensional rotations.

The interpreter's main requirement is that it be able to move around the datastructures it instantiates in a fluent and agile manner. This is necessary because the computational resources in the image-space processor are limited to representing at most two vectors at once, whereas the range of questions one needs to be able to ask of the whole system is large. For example, in order to answer the question "In which direction is the horse pointing?", the image-space processor has to be bound to the horse 3-D model in a completely different manner from that required to answer "Where is its front left hoof pointing?" at a particular instant during a step. Several interesting issues were brought into focus by having to design a satisfactory implementation of the interpreter.

The final component of the theory is something we call the *relaxation hypothesis*. This hypothesis states that by using the various cues available from the image, including information about obscuration, lighting and support as well as the lengths and angles observed there, it is usually possible to align the image-space representation of a viewed object accurately with the object's real-world orientation. Furthermore, this may be accomplished by a relaxation technique; that is, at any instant the compensating rotation is made about that axis (of the four available in the image-space processor) which reduces the largest discrepancy currently measured. We conjecture that this strategy will converge. This component is still a hypothesis because we have not yet finished implementing it. Our expectation is, however, that its implementation will strongly resemble that of the earlier visual processes with which we have had experience - i.e. it will consist of a considerable number of specialized diagnostics that interact in a fairly simple way.

It will be evident that the image-space processor is inherently powerful enough to represent two-dimensional patterns, such as the configuration of features on a face. Such patterns may be thought of as degenerate cases in which the girdle-angle is zero. The only requirement is that these patterns be described in an appropriate way in the database. This means that axes have to be set up in the two-dimensional pattern, and the configurations have to be described in the usual way relative to those axes. Interestingly, it has long been known that the choice of an axis in an image can greatly influence the way in which shapes are described. Figure 5 (Minsky & Papert 1972) shows an example of this. Important medium-level vision modules like symmetry-finding (Marr 1976) can be thought of as helping to find the axes that it is appropriate to use.

Image-space processor

We begin the detailed account of the computational facilities attached to each part of the theory by discussing the image-space processor. The interest here lies in minimizing the computational power that one uses. We require that the processor maintains (or simulates the maintenance of) six directions. They are:

(D1) $SPASAR, which in the minimal implementation is the only vector that can be rotated.

(D2) $AXIS, which is the vector to which the $spasar is attached and around which it rotates.

These two vectors are maintained in a local space-frame, which may be thought of as being defined by three directions:

(D3) $UP, which initially coincides with the gravitational vertical;

(D4) $FRONT, e.g. for a horse, the direction in which it is pointing, and

(D5) $HORIZONTAL, which is perpendicular to $up and $front.

Finally, because it is an important direction, we need

(D6) $VERTICAL, which is defined by the gravitational vertical.

The instruction set to the processor divides into four parts.

(P1) $spasar and $axis operations

(a) The $axis initially coincides with either $up or $front, (e.g. $up for a man, $front for a horse) and can be reset to these direction holders at any time.

(b) The $spasar can be attached to the $axis at a specified point and rotated around it. The most important three-dimensional relationship between the $axis and $spasar is called an *adjunct relation*, and it is written *(p i g)*. *p* is the position on the $axis at which the $spasar is attached; *i* is the *inclination* of the $spasar to the $axis, measured in the plane that contains them both; and *g*, the *girdle-angle*, describes the rotation of the $spasar around the $axis (see figure 7). The $spasar can also be translated away from the $axis according to certain rules. This makes it possible to represent the fact that one's arms are not attached directly to the axis of one's torso, but are translated away from that axis. This translation is carried out by means of an *embedding* relation *(d e)*, where *d* is the distance and *e* the girdleangle shown in figure 8. In the datastructures exhibited later in the article, adjunct and embedding relations are combined in one expression, which has the form *(p (i g) (d e))*.

(c) The $axis can be rebound to whatever the $spasar is currently bound to, and in so doing it assumes the spatial coordinates of the $spasar.

(P2) Space-frame operations

The space-frame may be rotated about any of the four directions $up, $front, $horizontal and $vertical (see figure 6). These operations are called respectively TWIRL, SPIN, TILT and VROTATE. In a minimal implementation, which is interesting for reasons we shall discuss later, executing these rotations would use the same machinery that rotates the $spasar about the $axis. Hence executing a space-frame operation would cause the current $spasar to be lost, having to be reconstructed after the frame transformation. If the $axis is not aligned with an axis of the space-frame, it too will have to be reconstructed. Because of

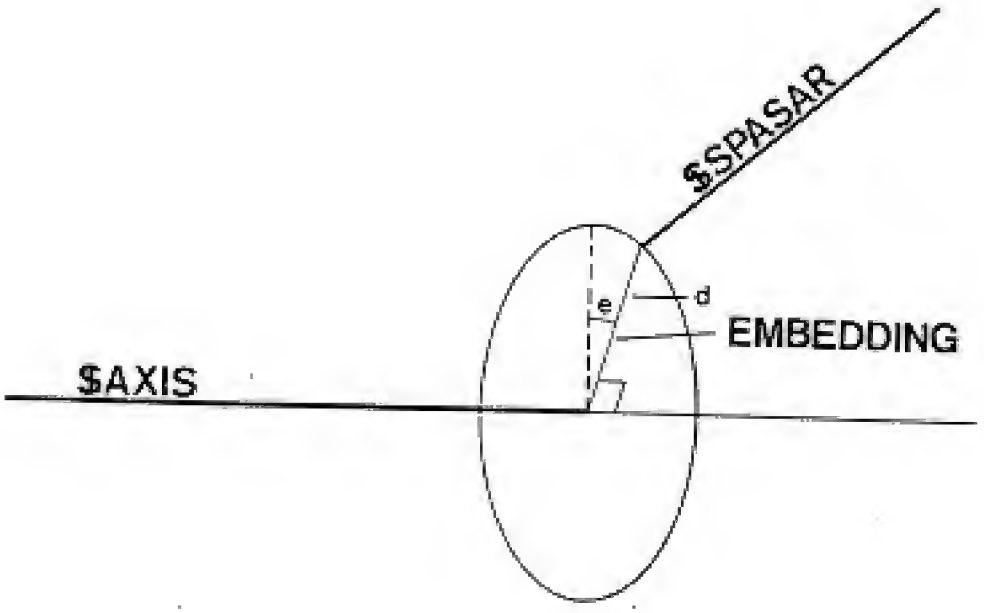


FIGURE 8. In practice, it is inconvenient to use an adjunct relation to describe the place at which an arm is attached to a torso. To describe this, and for example to represent places within a solid cylinder, we use an *embedding* relation (d, e), where d is the embedding distance, and e is the girdle-angle of the perpendicular from the embedded point to the axis. Adjunct and embedding relations are usually combined, and together they take the form $(p, (i, g)\,(d, e))$.

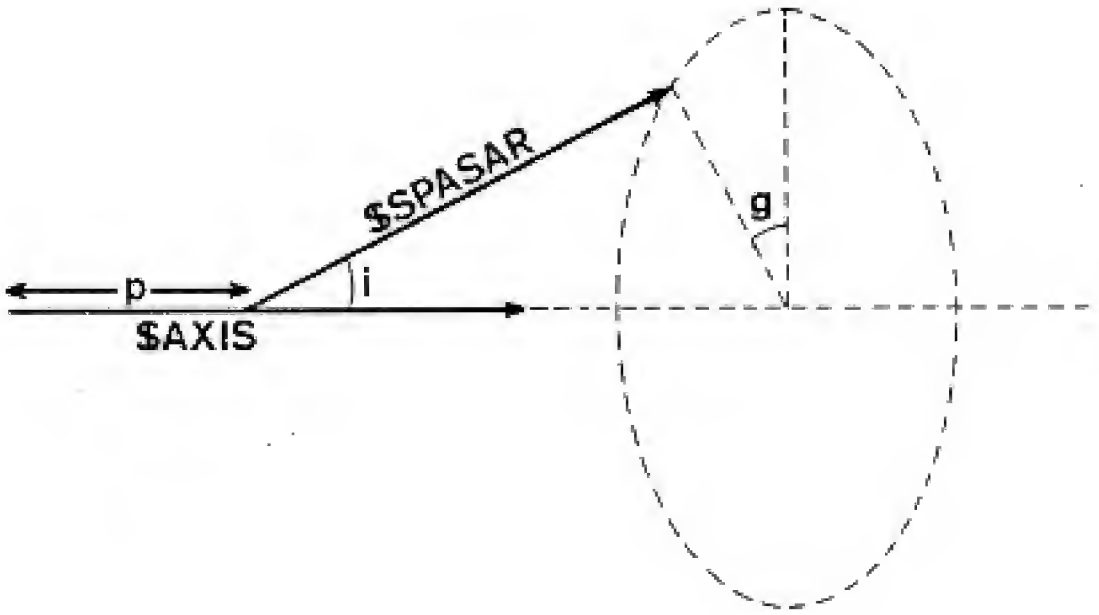


FIGURE 7. The most important 3-D relationship is called an adjunct relation, (p, i, g). The position p, the inclination i and the girdle-angle g are defined in the text, and illustrated here.

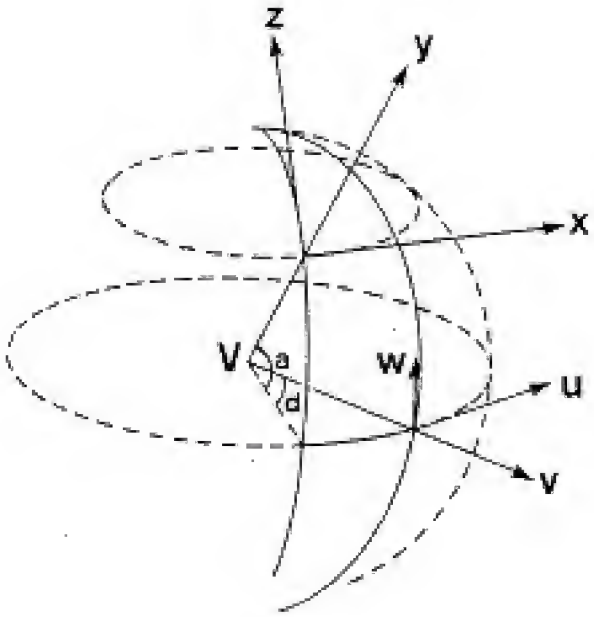


FIGURE 9. The orientation of the internal coordinate system depends on the angle of gaze. In order to minimize the computational power required to compute projections in the image plane, a vector's component towards the viewer is represented separately from its component in the tangent plane. As the angle of gaze changes, the coordinate system is rotated relative to the outside world. If the viewer V looks straight ahead, the internal coordinate system used is shown in the figure as (u, v, w). If the viewer moves his gaze to ascension a and declination d, the coordinate system is rotated $-d$ about the w axis, and $-a$ about the new direction of the u axis, resulting in the frame (x, y, z) illustrated here. In this way, no extra computation is needed at run time to determine the projection of a vector in the image plane. In order for this to be possible, the image-space processor requires accurate information about the direction of the angle of gaze relative to the gravitational vertical.

this, it is nearly always advisable to set up a new space-frame when the Saxis is rebound to a new part of the 3-D model. It is worth emphasizing that in this theory, information about the current direction of the gravitational vertical relative to the viewer plays an important role in defining the internal representation of the spatial disposition of a 3-D model.

(P3) Computing projections

In order to compute the appearance of a vector, one has to know the viewing angle. The natural geometry associated with an imaging system is spherical. When the image-space processor computes the lengths and angles associated with the Saxis and Sspasar, it therefore has to take account of the direction of gaze. This can be accomplished in two ways; either an initializing pair of rotations is made, about the Sup and about the Sfront, equal and opposite to the ascension and declination of the viewing angle; or this transformation is carried out on each vector just before its projection is read off.

The first method is simpler. It has the virtue that if the space-frame is actually represented in an internal coordinate system that specifies the component in the direction towards the viewer independently of the component in the tangent plane (see figure 9), the required projections are available without extra computation. Our present system is implemented this way.

(P4) Translation

After several changes of Saxis and Sspasar, one can find that the vectors are being constructed a considerable distance away from the origin of the image-space. This is not a problem for a computer implementation of the theory, but it would have to be considered carefully if the theory were construed as a psychological model.

The database and its interpreter

The overall picture of the datastructure that is set up for a particular image is of twenty or thirty independent atoms, each specializing in the description of one aspect of the image. For example, atoms might be set up for the head, torso, tail, forelegs, left-foreleg, hock, left-front-hoof, horseshoe, neck, mane, and pair-of-ears. Other atoms describe the shape of these items; for example, the tail may acquire the description "stick", the torso, a "cylinder". Special atoms will describe the texture of the tail, the colours of the various parts (the specific colours that occurred here), and the sheen on the animal's coat.

Each of these atoms names a particular type of datastructure, and contains on its property list (or the default extension of its property-list) values and relations appropriate to that type. For example, a torso-axis-atom has adjunct relations with the axes that connect to it, and a pointer to an atom for the torso's shape. The shape atom, "cylinder-shape", specifies the length and width of the cylinder, and points to any modifiers it may have - like bumps on it, whether it is flattened, and a description of the direction and degree to which it is conical.

After scrutinizing an image, it will be evident that the resulting datastructure can become very large. Finding the required information in it - i.e. evaluating a reference

within it - is therefore a major problem. We approached the problem by designing a system in which freedom of reference is a basic system facility, and by decentralizing as far as possible the indexes that support this freedom. For example, the top-level HORSE atom may be accessed directly if the system is asked to evaluate the reference $horse; but the pointers to the parts of that horse are kept in a subsidiary index at the particular horse atom. Thus in order to discover the atom that stands for this horse's tail, the atom for the horse must be interrogated. The horse-atom therefore acts like a local index - a management function that is superimposed upon its data-storage function - and we call such a local index a *packet*. Being a packet is mandatory for physical objects (PHYSOBs), but optional for lesser structures. Any atom can however become a packet, and it does so precisely whenever the viewer's interest in the image is sustained long enough for the pieces of description that lie "below" that atom to become instantiated. It is also possible to create a packet that organizes in a new way parts of a description that have already been instantiated.

The symbol-mapping problems

In order to construct mechanisms that would allow these things to happen, we had to design solutions to a complex of problems that have come to be known as *symbol-mapping* (Fahlman 1975, McDermott 1975). When he first introduced the term, Fahlman meant the phenomenon whereby being told that Clyde is an elephant causes much general knowledge about elephants to apply specifically to Clyde. Because much of the rest of Fahlman's discussion concerned recognition, there has been some confusion about the relevance of this phenomenon to recognition. There is no *a priori* reason why the two should be connected at all. There are in fact four quite distinct problems involved (Marr 1975b), and they are all important for systems that use stored descriptions to account for new data. The four problems are:

(a) The one Fahlman originally addressed, namely the application of general knowledge about elephants to the specific instance Clyde, which occurs when it becomes known that Clyde is an elephant. We shall refer to this as the *property-inheritance* problem, because here the task is to map properties held in the database onto a specific instance of a known class.

(b) Suppose that Clyde the elephant has already been mentioned in the current environment. Given only the reference ANIMAL, or LARGE GRAY OBJECT, or PEANUT-EATER, use this to *find* Clyde in the database. We shall call this the *reference-window* problem, since in some sense the issue is how wide to make the window through which you can access a current item by describing its properties.

(c) The reference-window problem merges into the third problem, that of *indexing for recognition*, but there is a distinction that is probably worth preserving. In the reference-window problem, the items that are to be referenced are already present as instances in the current environment. In recognition, the problem is to access a suitable template from the database with which to describe some incoming information. Recognition *results* in an instance of some template, whose own reference-window problem then begins.

(d) The reference-window problem meets the recognition problem somewhere near where they both turn into the fourth problem, the *problem of recall*. The difference between the

EXAMPLES OF $ATOMS

```
$1776
  torso $1676
  head  $1677
  leg:1 $1678
  leg:2 $1679
  adjuncts ((#1676 0 (0.0.0.0) nil))
  $class MONKEY
```

```
$1676
  adjuncts ((#1677  7 (0.0 0.0) nil)
            (#1678  0 (160.0 0.0) (5  90.0))
            (#1679 0 (350.0 0.0) (5 -90.0)))
  physob $1776
  $class MONKEY-TORSO
```

EXAMPLES OF TEMPLATES

```
MONKEY
  instances (#1776)
  head $MONKEY-HEAD
  arm:1 $MONKEY-ARM
  arm:2 $MONKEY-ARM
  torso $MONKEY-TORSO
  leg:1 $MONKEY-LEG
  leg:2 $MONKEY-LEG
  tail $MONKEY-TAIL
  parts (#head $arm:1 $arm:2 $torso
         $leg:1 $leg:2 $tail)
  adjuncts ((#torso 0 (0.0 0.0) nil))
  physob true
  direction up
  size 200
    .
    .
    .
```

```
MONKEY-TORSO
  adjuncts ((#head  7 (0.0 0.0) nil)
            (#arm:1 7 (135.0  60.0) nil)
            (#arm:2 7 (135.0 -60.0) nil)
            (#leg:1 0 (160.0 0.0) (5  90.0))
            (#leg:2 0 (160.0 0.0) (5 -90.0))
            (#tail  0 (135.0 180.0) nil))
  physob $MONKEY
  direction up
  size 100
  shape $TORSO-SHAPE
    .
    .
    .
```

FIGURE 10. Our internal representation of 3-D shapes is maintained on the property-lists of two kinds of atoms called templates and $atoms. Templates (upper case names like MONKEY) store information about archetype shapes while $atoms (names of the form $1776) are used to represent particular instances of the templates. A third kind of name is the $reference (template names prefixed with a $) which appear as values of properties in the templates above. These are decoupled pointers that indicate that one should first see if the particular reference has been instantiated in the current environment before following through to the indicated template.

(A) THE ENVIRONMENT BEFORE $EVALUATING "(:$THICKNESS $1876)"

```
$atoms:      $1876
               $class TORSO
                 .
                 .
                 .

templates:   TORSO                  THICKNESS              TORSO-SHAPE
               shape $TORSO-SHAPE     packets ($SHAPE)       thickness 200
                 .                      .                      .
                 .                      .                      .
                 .                      .                      .
```

(B) THE STEPS IN $EVALUATING THE REFERENCE (:$THICKNESS $1876)

```
(:$thickness $1876)          =>  look up thickness on $1876 property list
                                   its not there, so
                                 look up thickness on TORSO property list
                                   its not there, so
                                 look up possible packets of thickness in THICKNESS
                                   find $SHAPE, so evaluate:

(:$thickness $shape $1876)   =>  look up shape on $1876 property list
                                   its not there, so
                                 look up shape on TORSO property list
                                   find $TORSO-SHAPE, so instantiate it and continue:

(:$thickness $1976)          =>  look up thickness on $1976 property list
                                   its not there, so
                                 look up thickness on TORSO-SHAPE property list
                                   find it, and return its value: 200.
```

(C) THE ENVIRONMENT AFTER THE $EVALUATION.

```
$atoms:      $1876                                          $1976
               shape $1976                                    $class TORSO-SHAPE
               $class TORSO                                     .
                 .                                              .
                 .                                              .

templates:   TORSO                  THICKNESS              TORSO-SHAPE
               shape $TORSO-SHAPE     packets ($SHAPE)       thickness 200
                 .                      .                      .
                 .                      .                      .
                 .                      .                      .
```

FIGURE 11. This figure illustrates the processes involved in asking for the thickness of the torso, $1876. Information that will be used to answer this request is shown in (1) which is part of the environment at the time of the request. (2) shows the steps taken to answer the request, and (3) shows the effect this process has on the environment.

recall problem and the reference-window problem is that in the recall problem, the item to be accessed has not yet been brought into the current environment. The difference between it and the recognition problem is that one is not simply recognizing yonder ponderous gray object as *an* elephant; it is a specific elephant, namely Clyde, the one who ate your bag of peanuts last Thursday.

Some definitions

Only problems (a) (b) and (c) will concern us in this article, but before we can explain our solutions to them, we need to introduce the following definitions:

(1) *Template*

A template, denoted by an upper case name like HORSE, is an archetype property-list. Values in this property-list are expressed as $references (definition (3) below), not as particular instances.

(2) *$atom*

A $atom, e.g., $1776, represents an instance of a template, and is created by a process called *instantiation*. A $atom has a property-list, and the template from which a $atom is derived is signalled by the $CLASS property of the $atom. Property names prefixed by a $, like $CLASS, cause their values to be quoted rather than evaluated. It is roughly true that values not specified on the $atom's property-list default to values on the $atom's template (Raphael 1968 p. 85), but what actually happens differs in an important way from a simple default. We explain this below.

(3) *$reference*

A $reference is a name, prefixed with a $, which may be evaluated (by $EVAL) in the current environment. A $reference evaluation is successful if it returns a $atom. Typical $references are $ANIMAL, $HORSE, $SHAPE, $COLOR. References of the form ($reference1 . $reference2) are also allowed. In such cases, the result of evaluating $reference2 helps to define the context in which $reference1 should be evaluated (e.g., ($torso . $horse) or ($bump . ($neck . $horse))).

(4) *Packet*

A packet is a $atom that contains a local index, usually for some of the substructures of the object that the $atom represents. For example, if $1776 has $CLASS HORSE, one of the entries in its index might be (TORSO $1876), meaning that the torso of this particular horse was represented at the top level by the $atom $1876. Any $atom can become a packet if the items that it organizes are instantiated.

(5) *$index*

The $index is a general reference index. One can address it with an evaluable reference, and if an entry exists it will return another reference which will help in evaluating

the original. The distinction between the $index and the local packet indexes is that the $index is a permanent body of general knowledge (like a dictionary), whereas each local packet is always specific, and usually of only temporary importance. Because the $index consists of a set of small facts, it is quite easy to add to it; and because the $index is permanent, it can be allowed to grow very elaborate. At present, we keep $index entries under the templates they concern. A $atom representing a specific instance may be thought of as being plugged in to the general knowledge contained in the $index. The plugs are the entries in the $atom's organizing super-packets, because it is through these that a reference returned by the $index can evaluate to that $atom.

(6) *$EVAL, the reference evaluator*

When a $reference is made, it is evaluated by the function $EVAL, which is designed in accordance with the principle of least commitment. If a reference is made within a context that makes the referent unique, that referent is returned. For example, if an item on the $torso atom of a horse refers to $tail, the reference evaluates to *that horse's* tail. This evaluation would be unaffected by the presence of an ox elsewhere in the image. If however the reference was made externally, (for example by part of the datastructure for a soup recipe), the $tail reference would evaluate to an ambiguous pair. Other knowledge would have to be deployed to resolve the ambiguity. $EVAL makes extensive use of the information in the current environment and in the $index.

Examples of the datastructures defined here appear in figure 10.

The property-inheritance problem

We stated above that values that are not specified by the property-list of a $atom default to the values on the property-list of the $atom's template. What actually happens differs in an important way from this. The values that occur in the property-list of a template are (or contain) *$references*, not specific $atoms. $references cannot be used other than as references, they must be replaced by specific instances -ie by $atoms, before the associated template's property-list can be examined. Hence if a default to a $reference occurs, the interpreter evaluates the reference it finds, and places the result in the property-list of the $atom. If the reference has no referent in the current environment, a suitable one is instantiated.

An example will help to clarify this. Suppose that $1876 has $CLASS TORSO, and that the SHAPE held under the template TORSO is the reference $TORSO-SHAPE. When we evaluate the SHAPE property of $1876, the evaluator falls through to the template and encounters this $reference. If a $atom for this torso's shape already existed, it would have been indexed under $1876, so a new $atom ($1976 say) is instantiated from the TORSO-SHAPE template, and $1976 is entered as the SHAPE property of $1876. $1976 stands for the particular shape of this particular torso, although as it is now, $1976 consists mostly of defaults through to its template. The crucial ideas here are (a) that one can discuss only instantiated items and not references, so an unsuccessful reference evaluation results in an instantiation; and (b) that $1976 must be indexed in the $atom $1876 under the entry SHAPE,

not TORSO-SHAPE (although it may be indexed under TORSO-SHAPE as well). This is so that other $atoms can refer to it through the reference $SHAPE; they do not have to know that it is a particular TORSO-SHAPE. One important way in which templates can grow more specialized is by replacing general references like SHAPE with particular ones (like TORSO-SHAPE), but this is a much later issue. The act of instantiation is a common one, and tens of $atoms are created to describe even a simple image. The abundance of $atoms is one reason for the importance of the reference-window problem.

But suppose one asks for some property of $1876 that is not specified on the TORSO template, but which is specified somewhere. For example, we might ask what is the thickness of the torso? This information will not be found under TORSO or $1876 because it concerns the SHAPE of the TORSO, and so is one step removed from information that is directly accessible through $1876. This is the difficult part of the property-inheritance problem, and it was not addressed by Raphael (1968). Rather than trying to design a universal solution to this problem like Fahlman (1975), we took the view that the only thing needed to solve it is knowledge about where the required information may be found. This knowledge is held in the $index, as a specification of which packets can organize (in this example) THICKNESS. The $index returns the reference $SHAPE, which means that in order to discover the thickness of $1876 we have to interrogate its SHAPE $atom. If this exists, it will already be indexed under $1876's SHAPE entry. If it does not already exist, one is instantiated. This causes a second call to the $index to discover whether a special shape template exists for the shape of a torso. It does, and the $index returns TORSO-SHAPE, which is the specialist's internal name. This is instantiated, and the required information is then found. The important points here are (a) the use of the $index to guide the search round the database (like PLANNER theorems); and (b) once again, that using information from the $index involves a $reference evaluation which, if unsuccessful, can cause an instantiation.

The reference-window problem

The reference-window problem is important because the mechanisms involved here are what make it possible to apply one template (or scenario) to many different instances. For example, a HORSE template will contain references to $TORSO, which in any instance of that template need to evaluate to the particular $atom for that horse's template. The scenario "VIRGIN SACRIFICED TO FEARSOME-THING" will have internal references like $VIRGIN, $FEARSOME-THING, in terms of which information in the scenario - the motive of appeasement, and the secluded ingestion of one party by the other - is expressed. In a particular instance, the general statements contained in the scenario must be transformed into specific assertions about Mary-Jane and Godzilla; $VIRGIN must evaluate to Mary-Jane, and $FEARSOME-THING must evaluate to Godzilla.

The reference-window problem is almost the inverse of the property-inheritance problem, and to implement it requires extra indexing. To solve it we (a) decentralized the necessary indexing by distributing it among existing $atoms (this is what makes them into packets); (b) added entries to the $index that help one find the local index appropriate for a given reference; and (c) added extra top-level access-points where they

proved useful. We have already seen several examples of (a) and (b), and an example of (c) is that when a HORSE template is instantiated, its $atom is attached to the $index by the top-level references $HORSE and $ANIMAL. The reason for this is that much knowledge about a HORSE is best represented as knowledge about ANIMALs, for reasons of economy.

How packets are created

As we mentioned earlier, the 3-D representations that we use are loosely hierarchical. The hierarchy (such as there is) is expressed by the local indexing structure, and it comes about because $atoms can often cause the instantiation of other $atoms that they then proceed to index. This can happen in several ways. One is to ask the $index for a list of the PARTS of a 3-D model. It will return a list of templates - the PARTS of an ANIMAL consist of its HEAD, NECK, TORSO, FORELEGS, HINDLEGS and TAIL. The process of ACTIVATION is defined as the instantiation of the PARTS of a $atom. It occurs whenever something important happens to that $atom - for example if the $atom is used to set up a space-frame in the image-space processor - and it may be thought of as the lazy man's way of instantiating a 3-D model.

The instantiation sequence is not restricted to using the PARTS list; in a special circumstance, the processor can avoid the PARTS recipe altogether, instantiating only what is required by current needs for example by spatially driven index accesses to what is at the front of the torso or its rear, etc. Thus it can happen that different parts of an animal are currently represented at quite different levels of detail. If the front left hoof is being scrutinized, there is no reason why the hindlegs should have been unpacked beyond a coarse HINDLEGS descriptor. The hierarchy in 3-D model organization is not strict, and the local ordering is not even total. For example, one can move straight from the torso to the left foreleg; it is not necessary to pass through the pair-of-forelegs datastructure. The particular route that one chooses and the amount of detail for which one instantiates descriptors depends upon the purpose for which the animal is being viewed.

The reverse of activation also occurs; that is, instantiating a given $atom can cause the instantiation of a superior one that then indexes the original. This is so important that it always occurs unless a suitable superior already exists, or unless the $atom is itself a PHYSOB (which has a top-level status). Indexing is important because if an item is not plugged into the $index somewhere, all references to it will fail, and the item will essentially have been lost forever. For example, if a TAIL template is instantiated, it will cause the instantiation of an ANIMAL $atom which then attaches itself to the $index, and through which references like $TAIL or ($TAIL $ANIMAL) may be evaluated successfully. In this way, a particular tail becomes connected to the general knowledge about animals and animal tails that is held in the $index. It is interesting how a strong physical property like cohesiveness comes to be reflected in the datastructure as an indexing strategy.

Interfacing with the image-space

Before we discuss the third symbol-mapping problem, that of indexing for

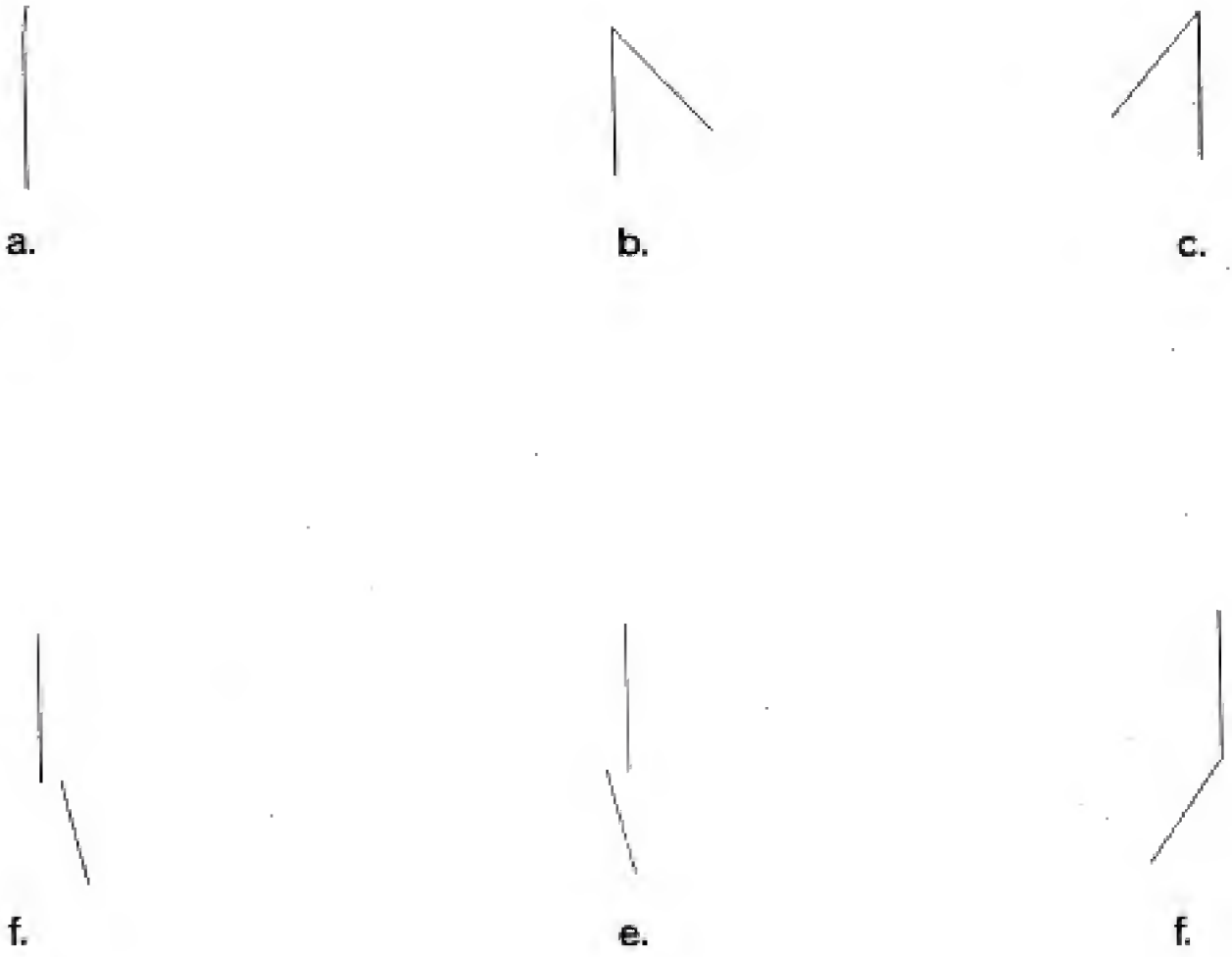


FIGURE 12. The $axis and $spasar are used to interpret an adjunct relation within a space frame. In the figures above, the $axis is maintained on the monkey's torso while the $spasar is moved onto the various adjuncts of the torso. In (a), the $spasar is bound to the axis of the monkey's head. In (b) and (c), it is bound to axes for each arm; in (c) and (d), to axes for the legs, and in (e), to the tail. We see how each pair would appear from a particular orientation.

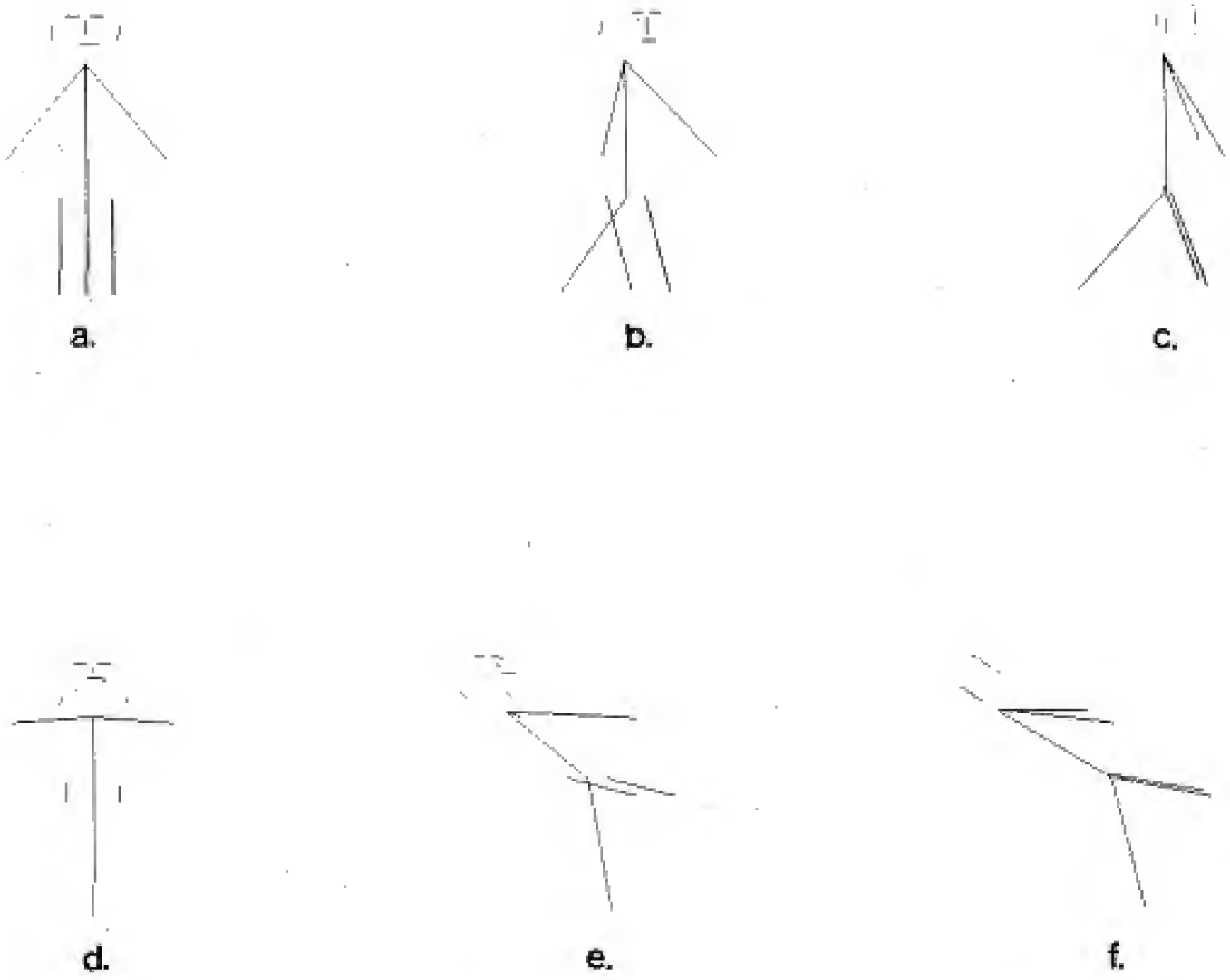


FIGURE 13. The monkey shown in several space frame orientations. Each image is produced by mapping out all of the monkey's instantiated axes while keeping the space frame fixed. For example the monkey's nose is found by starting with the $axis on the monkey axis and locating the torso axis with the $spasar, then the $axis can be set to the $spasar's position, fixing it on the torso, and the neck can be located by the $spasar using the torso-neck adjunct relation. This leap frog process continues until the $spasar is on the monkey's nose. When the space frame is rotated, we must reconstruct the entire image. An important corollary to this is that we only construct those details we need and those only when they are needed. In (a), (b) and (c) a standing monkey is rotated about $vertical in 45 degree steps. In (d), (e) and (f) the monkey is first tilted backward 45 degrees and then rotated as above.

recognition, we need an understanding of how the pieces of machinery described so far work together. Let us therefore assume that a 3-D model has already been selected, and observe how it interfaces with the image-space processor.

Suppose that the commands (INSTANTIATE MONKEY) and (ENTER-IMAGE-SPACE $MONKEY) are executed. The effect is to instantiate a $atom from the monkey template, to activate it (which instantiates the major parts of a monkey), and to set up a space-frame for the monkey. The $axis is bound to the monkey-atom, and the $spasar is not yet assigned.

By executing (CHANGE-$AXIS $TORSO), and then interpreting the adjunct relations between the torso and the animal's limbs, the $spasar may be placed in any of the positions shown in figure 12. By rotating the space-frame about the four allowed axes, the stick-monkey may be rotated to any orientation (figure 13).

It is important to remember that in practise, only two vectors (e.g. the $torso and the $tail) are represented at any one moment. When the $spasar is rebound to another limb, record of the predicted appearance of the previous one disappears, and only the adjunct relation that was read off the $spasar remains in the 3-D model. The IMAGEL property of the tail's $atom will still point to the image element in question, and if this moves, flags will have to be set to warn that the former adjunct relation may now be invalid. We can reasonably suppose that in real life, IMAGEL bindings into a 3-D model can be maintained even while the $spasar is bound elsewhere, because relatively low-level tracking algorithms suffice to follow a moving item in an image (Chien & Jones 1975, Speckert 1975).

Indexing for recognition

We have seen how the theory's mechanisms run in an isolated state, and we turn now to the relation between those mechanisms and the image during recognition. The interesting point raised by pipe-cleaner animals (figure 1) in the context of the present theory is that one has to use a 3-D description from the database before the image-space processor can be run: but one might think that once a 3-D description has been selected, recognition has in some sense already taken place. In a full image, one might argue that "other clues" suffice to select the appropriate 3-D model, but in the pipe-cleaner model there are no other clues. The traditional A.I. answer to this dilemna is to hypothesize and test, using error information in some kind of "difference-directed memory" to move to a better hypothesis (Minsky (Fahlman) 1975). This strategy violates the principle of least commitment on which much of our vision system is built. Fortunately, it is possible even at this late stage to design the diagnostic system so that it obeys the principle of least commitment. This may be accomplished by using a "general" animal 3-D model (possibly several - large, medium and small-animal), whose axes are only roughly correct, and which either has no functional semantics pointer itself, or only a weak one. Using a general model, good estimates can usually be made of the actual lengths, inclinations and girdle-angles present. By accessing the index using this new information, the correct 3-D model and its functional semantics pointer can then be recovered. When the correct 3-D model is eventially found, the only visible change to the top-level organizing $atom occurs in the value of its $CLASS property. In some sense this change constitutes the act of recognition, because after it has occurred, references

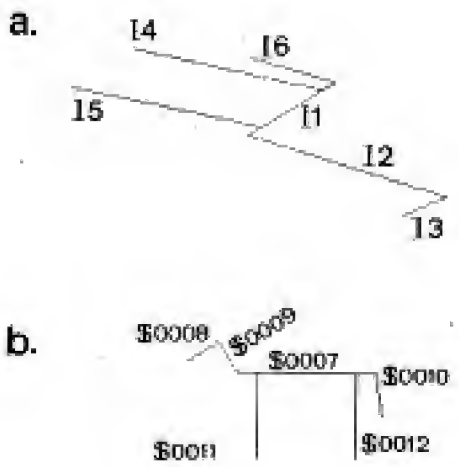
will automatically evaluate into the "recognized" template.

Figure 2 showed a typical datastructure that might have been delivered by the lower parts of the vision system. It consists of a set of axes, computed by segmenting a form (Marr & Vatan, to appear), bound to each of which is a property list (possibly null). This property-list may for example describe the shape of the generalized cylinder whose axis this is, and possibly some descriptors of its surface texture. In a pipe-cleaner animal these properties are irrelevant. The first problem is to access an appropriate 3-D model from the database. There are various parameters by which these models may be indexed. Connectivity is not destroyed by perspective transformations, nor are numbers like the fractional distance down one axis at which another axis connects to it. Spurious connectivities can of course be introduced if one axis crosses in front of another, and if the reason is not recognised lower down; but existing connections cannot be destroyed, only obscured. Hence in order to use the connectivity information, when measuring which database items best match a given configuration set, unexplained errors of omission are treated much more seriously than unexplained errors of commission.

The second sort of information is girdle-angles, inclinations, and the relative lengths of axes. It is easier to take advantage of these later on, when the image-space processor has delivered at least partial results about the three-dimensional orientation relative to the viewer; but it is possible to do something with them early on. This comes about through weak, gross clues. For example "verticals" in the image are often close to verticals in real life, and if the apparent length of a "neck" exceeds the apparent length of a leg, and if both are quite large, the image is likely to be a giraffe. In other words, lower bounds can often be inferred, and are sometimes useful. Another important type of clue concerns major differences in the girdle-angles of two axes that are connected to a common one. For example, the neck and the tail often point in very different directions – one up and one down – and this obvious difference can be seen without a sophisticated 3-D analysis. In a pipe-cleaner animal, this very rough difference can help to determine which end of the animal is which.

Bearing these considerations in mind, we see that indexing clues can be divided into two kinds; those that can be used before the image-space processor has been called into play, and finer clues that require at least a preliminary guess at the 3-D configuration before they become sufficiently reliable. The former category includes connectivity, fractional lengths on one axis, some comparisons between the lengths of different axes, very rough relative girdle-angle comparasons between two axes that connect to a common one, texture and rough shape information, and possibly general information like the number of axes that are probably horizontal, vertical, or neither. Such information in fact provides a surprisingly rich body with which to go to the indexer, and this is part of the reason for our opinion that straightforward recognition (recovery of the functional semantics pointer) is considerably over-determined in a natural system. The second category includes much finer indexing on the relative lengths of different axes and their 3-D relation to one another.

We can now follow the course of the analysis that takes place when the system is presented with the datastructure shown in figure 2. Firstly, the connectivity of the axes is




```
$0010
INCREMENT: (125.000005 74.0499511)
ORIGIN: (44.642862 26.44628351)
SHAPE: $CYLINDER
END2: (169.642868 100.4957941)
END1: (44.642863 26.44628351)
IMAGE-NAME: I1
IMAGEL: TRUE

$0003
INCREMENT: (287.28868 -82.4423761)
ORIGIN: (44.642862 26.44628351)
SHAPE: $CYLINDER
END2: (331.50955 -55.9361131)
END1: (44.642862 26.44628351)
IMAGE-NAME: I2
IMAGEL: TRUE

$0002
SHAPE: $BLOB
END2: (271.54021 -80.5588511)
END1: (331.98855 -55.9361131)
IMAGE-NAME: I3
IMAGEL: TRUE

$0009
INCREMENT: (-270.632848 53.4400551)
ORIGIN: (151.785722 80.067291)
SHAPE: NIL
END2: (-118.847229 143.3578441)
END1: (151.785722 80.067291)
IMAGE-NAME: I4
IMAGEL: TRUE

$0004
INCREMENT: (-270.632546 53.4400561)
ORIGIN: (62.580005 37.0247621)
SHAPE: NIL
END2: (-208.13234 90.4654211)
END1: (62.580005 37.0247621)
IMAGE-NAME: I5
IMAGEL: TRUE

$0005
INCREMENT: (-119.894452 34.3585661)
ORIGIN: (151.642866 100.4957941)
SHAPE: $STICK
END2: (45.948414 134.8467821)
END1: (169.642868 100.4957941)
IMAGE-NAME: I6
IMAGEL: TRUE

$0006
SPACE-FRAME: $0013
PRINCIPEL-AXIS: $0007
ACTIVATED: T
KINGLEGS: $0012
FORELEGS: $0011
TAIL: $0010
NECK: $0009
HEAD: $0008
TORSO: $0007
PACKET: TRUE
ANIMAL: TRUE
IMAGEL: NIL
$CLASS: $PANIMAL
PHYSOB: TRUE

$0007
PACKET: $0006
TORSO: TRUE
IMAGEL: $0000
$CLASS: $TORSO
RELGIROLES: ((NECK $HEAD -100.0))
REFERENCE-AXES: (CHECK $ANIMAL)
ADJUNCTS:
            ((#0005 0 (120.0 . 30.0))
             (#0002 G (90.0 . -90.0))
             (#0011 L (90.0 . -90.0))
             (#0012 0 (120.0 . 30.0))
             (#0010 J (90.0 . -90.0)))
SHAPE: $CYLINDER
PHYSOB: $0005

$0008
PACKET: $0006
HEAD: TRUE
IMAGEL: NIL
$CLASS: $HEAD
PHYSOB: $0006

$0009
PACKET: $0006
NECK: TRUE
IMAGEL: $0003
$CLASS: $NECK
PHYSOB: $0006

$0010
PACKET: $0006
TAIL: TRUE
IMAGEL: $0005
$CLASS: $TAIL
PHYSOB: $0006

$0011
PACKET: $0006
FORELEGS: TRUE
IMAGEL: $0004
$CLASS: $FORELEGS
PHYSOB: $0006

$0012
PACKET: $0006
HINDLEGS: TRUE
IMAGEL: $0002
$CLASS: $HINDLEGS
PHYSOB: $0006

$0013
PACKET: $0006
PHYSOB: $0006
$CLASS: $SPACE-FRAME
HORIZONTAL: (0.0 234.923217 -85.5050431)
UP: (0.0 85.5050375 234.923163)
FRONT: (250.0 0.0 0.0)
SPACE-FRAME: TRUE
```

FIGURE 14. The first step in the processing is to find a candidate 3D-model for the imagels shown in (a). We do this using information from the imagels that is least sensitive to distortion by rotation. This includes axis connectivity and relative distances along an axis, as well as parallel axes and gross differences in length. Once a 3D-model is selected, many of the imagels can be given a polarity, that is one of the ends of the imagel is distinguished as the origin of the imagel. In this example the 3D-model obtained by addressing the index with coarse clues is an ANIMAL. A Satom ($0006) was then instantiated from the ANIMAL template, it was activated, linked to the imagels, and entered into the image space in standard orientation. Notice that at this point, imagel I3 corresponding to the head has not yet been bound to its Satom. This is one of the next things to happen.

discovered, and coded ready for accessing the indexer. Nevatia (1974) also described accessing an index using such connectivity information. The connectivity and the distance down each axis at which the other connects to it are both used. Each join is characterized using an overlapping hash-bucket system that makes adequate allowance for errors of measurement. At the same time, the axis (or axes) with the most connections are noted, and they form a possibilities list for the principal axis of the structure. In this example, there is no ambiguity, because the torso axis has many more connections than any other.

The call to the indexer returns a general ANIMAL 3-D model, together with a list of several specific animals. In real life, it is likely that auxiliary information would suffice to narrow this set of possibilities down to a single candidate, but here we take the more difficult course of using the general ANIMAL model. This 3-D model is instantiated and activated, and the $atom for the torso is identified with the principal axis in the image. This identification is not yet complete, because we have yet to resolve its polarity (i.e., which end of the animal corresponds to the head, and which to the tail).

The polarity of the torso is over-determined in this example. It could be recovered from the tail-down and neck-up combination; or from the rough shape-descriptors that were happily included in the problem statement. Once the polarity of the torso is determined, the system proceeds to re-express the image elements in the form (origin, increment), rather than the form (end1, end2) that was originally given, because that is the form needed for matching with the image-space processor, and because it now has enough information to be able to do this correctly. The image elements are then bound into the IMAGEL properties of the $atoms for the remaining parts of the ANIMAL. This binding includes backpointers, so that (for example) were one leg to move and this fact noticed by low-level routines, such routines could interrupt the higher structures directly rather than by having to initiate a top-down search for the IMAGEL that changed.

The result of this process is the datastructure shown in figure 14. The system now instantiates a space-frame for the animal, and is ready to commence the relaxation process that will result in the representation of its three-dimensional disposition.

Unusual views

In real life, initial access to a suitable 3-D model may be more difficult than this, because the axes that emerge from segmenting a two-dimensional form can differ in an important way from the axes that are natural for the 3-D model. One circumstance in which this can happen is when an important axis of an object points directly towards the viewer. For example, the side view of a bucket segments naturally into a generalized cylinder description in which the bucket is represented as a slice of a cone, and the axis is vertical. If one looks at the bucket from above, one essentially sees two circles joined by the sloping sides. The principal axis of the bucket appears as a point from this perspective. The same phenomenon is exhibited by the image of a long, thin cone whose axis points nearly directly away from the viewer. Also, when a viewed object is very close, peculiar distortions can occur in the relative sizes in the image of its parts.

In order to access the correct 3-D model despite these obfuscations, some idea

of depth has to be introduced into the analysis *before* addressing the 3-D model index can be successful. In the case of the bucket example, some process has to realize that the two circles might be separated in depth, and that if they are, they could be separated by a considerable distance. The clues that signal this are often nuances of shadow and highlight, and this leads us to expect that much of the analysis of lighting and shadow can influence the processing at exactly this stage of recognition. We think of the computations that take place here as deploying the $spasar to construct from the image a primary 3-D model that consists at first of an axis in depth whose circumscribing surface is bounded by the two visible circles, and to which extra details - like hollowness, the closure of one end of this surface by an orthogonal plane, and possibly the addition of a cross-strut to account for the handle - are added. At some point during the construction of this description, the indexer is successful at finding a match with a bucket 3-D model in the database. We do not at present understand this any more precisely, but we have the feeling that one might have to abandon the principal of least commitment here in favour of some kind of hypothesize-and-test strategy. If an "unusual view" becomes a common view, it would become profitable to index the appropriate 3-D model under the special features that obtain for that view.

Interestingly, Warrington & Taylor (1973) found that patients with lesions in the right parietal lobe were greatly impaired when confronted with unusual views of objects. Such patients, who can recognize the picture of a bucket in side view, are unable to recognise the same bucket when viewed from above; and even deny that the latter could be a bucket when informed that it is. The authors commented that unconventional lighting was as effective as an unusual perspective in disturbing the performance of such patients, and they suggested that this arises because the more straightforward 2-D features are absent in these situations. These findings were recently confirmed by Carey (personal communication) who agreed that "unusual views" usually correspond to views where an important axis is foreshortened.

Relaxation

With the image elements properly bound to a 3-D model, we can begin the relaxation process. This is an incremental activity of adjusting the $axis' and $spasar's orientations, guided by constraints contributed by the image, the 3-D model, and external influences such as gravity. The object is to find the correct 3-D orientation which will allow us to calculate the true lengths of the axes and their relative dispositions in space. This information can then be used to access a more specific 3-D interpretation of the image.

The constraints we have to work with are varied, and we have been finding new ones quite regularly. Most of them contribute information that restricts the disposition of one axis relative to another, sometimes in rather complex ways. The major problem in the relaxation task seems to be in combining these incomplete clues to deduce the 3-D model's actual orientation. The image-space processor is currently the major focus in the relaxation theory that is developing around this problem. We feel that this processor can be used effectively as a dynamic model of space, where the $axis and $spasar are used much as a child plays with blocks to see how they can go together given the additional constraints he wants to impose on the configuration. In our case, the $axis and $spasar are held together as rigidly

as is necessary to maintain whatever adjunct information we have available, and the whole configuration is pushed around until its projection onto the image plane most closely matches the image we are trying to interpret. Knowledge of these constraints and of methods for applying them will be represented procedurally in the relaxation processor currently being developed. In the discussion that follows, we outline some of the orientation constraints available to the relaxation processor and how we plan to apply them.

The first constraints on an axis' orientation are calculated directly from the image. The imagel gives us the orientation and length of its projection onto the image plane, so the only additional information needed, to compute the axis' orientation, is the inclination this axis makes with the viewer's line of sight. Sometimes this angle can be estimated directly from the imagel's shape description. For example if the axis is known to be cylindrical and the intermediate visual processor has supplied a shape description sufficient to calculate the location of its perspective vanishing point, the axis will be parallel with a line from the viewer to this vanishing point. More often, the imagel's shape property and our confidence in what it should be will not allow more than a very approximate guess at the vanishing point's location, so other clues are needed to constrain the axis further.

The influence that gravity and other external factors have on the distribution of imagel orientations provides additional information about individual imagels. For example if an imagel is close to vertical in the image plane, there is a high probability that the corresponding axis is vertical in space as well. This clue does not require knowing anything about the nature of the observed axis to be used, but it attracts most confidence when the axis is known to be vertical in its normal state (e.g. a horse's leg). Another clue has to do with the ground. It is often flat, and when it supports the objects we view, it serves as a very strong constraint on the likely orientations of some of their axes. Consider for example a cow's torso; it is very difficult for the cow to hold it out of the ground plane without bending his legs out of parallel.

When individual imagels fail to provide enough information, we can look at several together. In the example with the cow, it was important that its legs were parallel for insuring that his torso was parallel to the ground. Parallelism is a very nice property in that the imagels that correspond to parallel axes are also parallel. In animals, the legs are quite often close to parallel and this is a powerful means of disambiguating them from other adjuncts of the torso. Accidental alignments can produce parallel imagels where the axes are not parallel, but rarely will this happen for more than two axes at a time, and even this possibility is infrequent enough to justify paying attention to any parallel axes that are present in the image.

When one axis has been fixed in space, the dispositions of its adjunct axes are heavily constrained and are often determined uniquely. It is a simple matter to rotate the Sspasar about the known axis until its projection matches the imagel of the unknown axis. This is what was done above when the animal's torso was rotated about the vertical until its projection matched the torso imagel.

Finally if no axis is determined but two adjunct relations are known between three nonparallel axes, then (except for a few pathological cases) the dispositions of these

axes are determined. The image-space processor can be used to discover these dispositions through a process of experimenting with various orientations of these three axes (two at a time) attempting to maintain the adjunct relations while minimizing the discrepency between the imagels and the projected axes.

Let us now return to the animal image we have been processing. It was left at the point where its 3-D model had been activated and entered into the image-space with the imagels correctly bound to its axes. It has no vertical axes, but it is known to be on the ground and its torso is very likely to be horizonal. The imagel shape information is not good enough to calculate any vanishing points, and only the adjuncts between the torso and the legs are reasonably certain since the others vary too much from one animal to another. First the Saxis is set to Svertical and the Sspasar is placed on the animal's torso adjunct. Thus the Sspasar is forced to be perpendicular to the gravitational vertical. The Sspasar's projection must be aligned with the torso imagel so that the tree end of the Sspasar is on the tail side of the imagel. Once this is done, the Sspasar is oriented as the animal's torso is. Next the animal 3-D model must be rotated about its torso axis so that it corresponds with the image. The torso-leg adjunct is the most reliable, so the Saxis is moved to the Sspasar's current position on the torso and the Sspasar is put onto one of the legs. Again the Sspasar is rotated about the Saxis until its projection onto the image plane matches the corresponding leg imagel. At this point the animal's orientation is determined. The next task is to measure the remaining adjunct relations. A very important constraint for doing this comes from the fact that an animal's axes tend to be parallel to one plane. This means that the girdle angles of the adjuncts are the same modulo 180 degrees. So the inclination of the animal's neck can be found by putting the Sspasar on the torso-neck adjunct and varying the inclination angle until the Sspasar's projection lines up with the neck imagel. Finally the relative lengths of the axes are measured by setting the Saxis and Sspasar onto two axes at a time and adjusting their lengths until they just cover the corresponding imagels. Figure 15 sketches the relaxation sequence described above.

Minimizing the complexity of the image-space processor

In our account of this theory, the image-space processor has been pared down to almost its minimal implementation; only six directions are represented, only two are active and (as a corollary) only four rotations allowed. The original motivation for this was to see how simple a processor could support the mechanisms that were required. Provided that coordinates are chosen as described in figure 9, the computations needed to support the image-space processor are straightforward, and it is unlikely that finding an economical neural representation for this part of the theory will prove very difficult.

The main characteristic of a minimal implementation is that it should contain only one "rotatable" element, and rather few passively stored directions. In such an implementation, a construct in the space-frame would not survive rotation of that frame, and would have to be reconstructed. This is quite a strong signature of a minimal implementation. The ability to hold directions in passive storage would sometimes be useful, though some care is necessary when deciding whether they are still reliable.

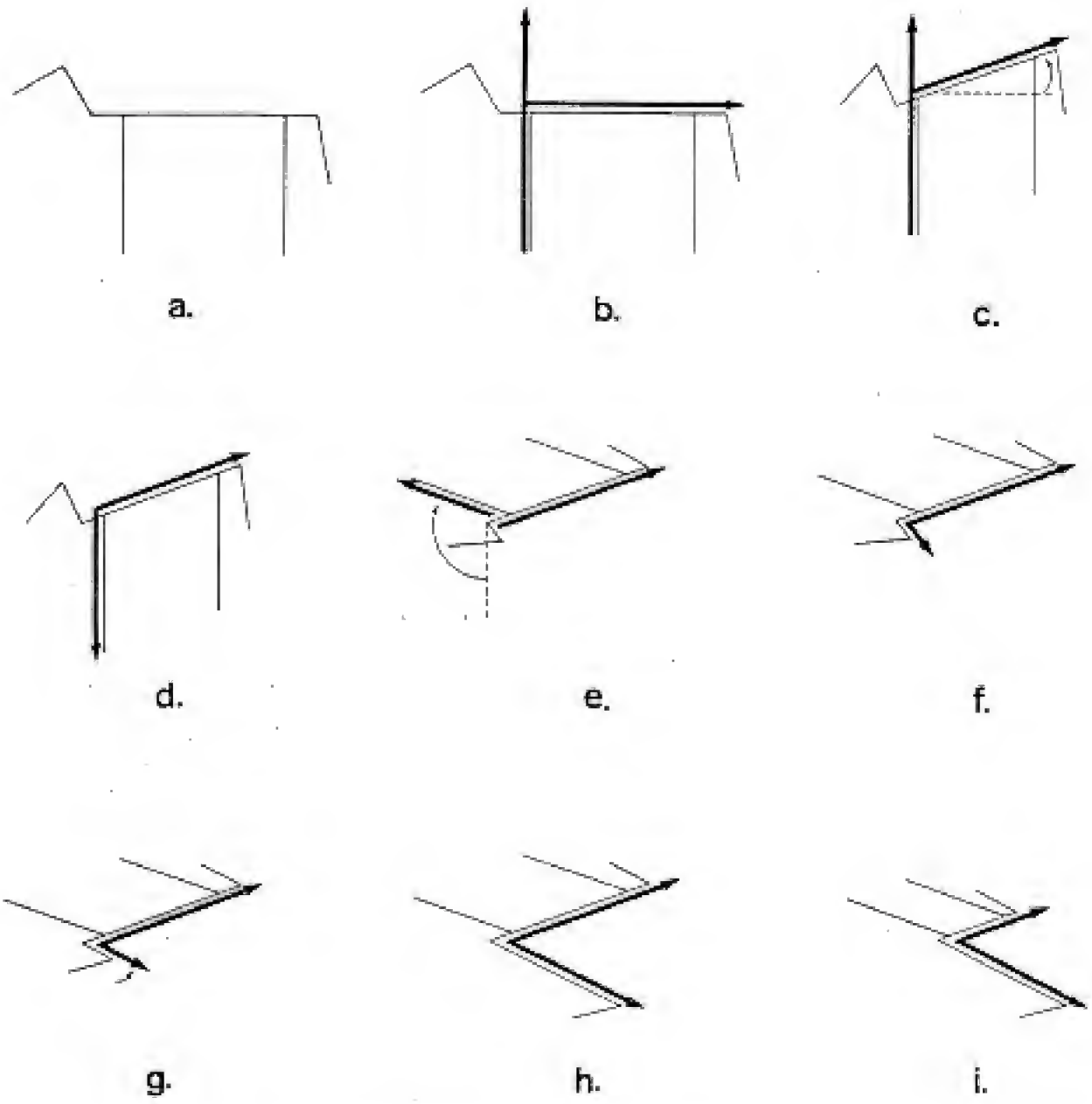


FIGURE 15. Once a 3D-model is selected for the object, it must be rotated to the disposition of the image so that the object's axis lengths and adjunct relations can be measured. This process is carried out on the example shown in figure 14. (A) shows the 3D-model in its standard orientation. To get the orientation of the torso, an assumption is made that it is parallel to the ground. The $axis is set to $vertical and the $spasar in (B) is set perpendicular to it in the image plane. The $spasar is then rotated about the $axis until its projection aligns with the torso image1 in (C) establishing the direction of the torso axis. Next the 3D-model's rotation about this axis must be discovered, so the $axis is moved to the $spasar's position on the torso and the $spasar is placed on the torso-leg adjunct and rotated about the $axis until its projection aligns with the leg image1 (D) and (E). At this point the 3D-model is correctly oriented. In (F) and (G) the inclination angle of the torso-neck adjunct relation is measured by placing the $spasar on the neck axis and adjusting the inclination until its projection aligns with the neck image1. In (H) and (I) the lengths of the torso and neck are compared by lengthing the $spasar until its projection is the same as the neck image1 and shortening the $axis to match the torso image1.

Otherwise, the structure of the theory turns less on the constraint of minimal complexity than one might at first sight expect. For example, one important characteristic of our representation is its ability to move fluently from a coarse one-axis description of a whole animal to a very fine description of one small part. By removing the minimality constraint, one could easily design a machine capable of maintaining a fine description of all parts of the animal simultaneously, but what would be the point? The fact is that for many purposes, a complete 3-D reconstruction of a physical object is no better than the original object. How do you answer in which direction a horse is pointing *unless* part of your representation contains something like a torso axis? Many of the diagnostics and constraints that apply during recognition and relaxation concern the overall disposition of the whole animal, not very local details (though they too can be important). The same is true of the questions that one asks of an image in real life. In a very luxurious implementation, one might maintain complete descriptions at all levels of detail simultaneously, but the resources needed to do so could probably be better employed in other ways. Provided that a simple implementation can compute answers to the important questions reasonably quickly, there is no reason to use a complex implementation.

Discussion

The discussion falls naturally into two parts, one concerning the specific 3-D representation theory, and the other dealing with the broader issues raised by the interpreter for that theory.

1: 3-D representation theory

There are five main points to our theory. They are:

(1) The 3-D disposition of an object is represented primarily by a stick-figure configuration, where each stick stands for one or more axes in the object's generalized cylinder representation.

(2) This configuration is described by a loosely hierarchical assertional database, called a 3-D model. Use of this database is extremely free and flexible, and it can support levels of description that cover the spectrum from very coarse to very fine detail. (N.B. the principle of graceful degradation.)

(3) In order to be useful, this database has to be interpreted through an (essentially) analogue mechanism, called the image-space processor. In its minimal implementation, this processor maintains a representation of two directions in a space-frame, in addition to the gravitational vertical.

(4) The image-space processor's instruction set is small. Its most important features are:
   (a) the ability to interpret an adjunct relation between the $axis and the $spasar; and
   (b) the ability to execute four frame rotations (about $up, $front, $horizontal and $vertical).

(5) The image-space processor can deliver information about the lengths and orientations of the appearance of the $axis and $spasar. These help the system to rotate its model into the correct 3-D disposition relative to the viewer.

It has not escaped our notice that this theory may illuminate certain recent findings in experimental psychology. Shepard & Metzler (1971) created a set of images by rotating and reflecting simple objects made of cubes (figure 16). They found that the time taken to decide whether two such images were of identical objects, rather than objects that differed by a reflection, varied linearly with the angle through which one object must be rotated in 3-space to become aligned with the other. This finding revived interest in "mental imagery" and in analogue processes in perception (Cooper & Shepard (1973), Metzler & Shepard (1974), Shepard (1975)). In addition, Kosslyn (1975) has published evidence for an analogue component to the processes that interpret mainly two-dimensional structures, like faces and maps.

The significance of such experiments is controversial (but not the results). It is a commonplace that an observer's description of what he does not understand is often misleading, because the concepts through which he attempts to capture the experience are inadequate. Because of this, "mentalist" experiments and especially the introspective reports that accompany them, are rightly regarded with suspicion. Although it is widely recognized that a complete theory of mental processes will eventually have to explain the findings and the subjective experiences that accompany them, we are probably not alone in feeling that one should not rely on their help to construct a theory, because of the possibility that such reports will become accurate only after the observer understands the processes that are giving rise to them.

Another reason for the controversy seems to have been the difficulty in seeing how an "analogue" process could benefit the computations that underlie perception and recognition. We believe that the present theory shows a way in which such a mechanism could be useful, although we recognize that this may not be *the* way in which we in fact use it. In order to help decide this, one probably has to study the neural implementation of the mechanisms that we described, in the hope of making testable predictions about single unit responses. Since this is a major undertaking, one needs some evidence that the theory is indeed a likely candidate. There are several points that appear to us to constitute reasonable grounds for believing the theory to be a good candidate for a psychological theory. They are:
(1) Pipe-cleaner animals are almost as easily recognizable as are line-drawings of animals, despite their very abstract relation to the original. This would not be surprising if pipe-cleaner animals were in some sense extracted from the image during the normal course of its interpretation (as our theory asserts), but it would be surprising if not. The computational advantages of so doing need no emphasis.
(2) The loosely hierarchical structure of our 3-D models has many computational advantages that are almost bound to be shared by the psychological representation, even if the psychological representations are otherwise very different. One can probably rule out any system that cannot restrict the level of descriptive detail at any point to only that currently needed.
(3) An important part of the theory is the minimal nature of the image-space processor. A consequence of this is that after executing a rotation, the "image" of the 3-D model has to be reconstructed in the new space-frame, as opposed to being constructed once and then being

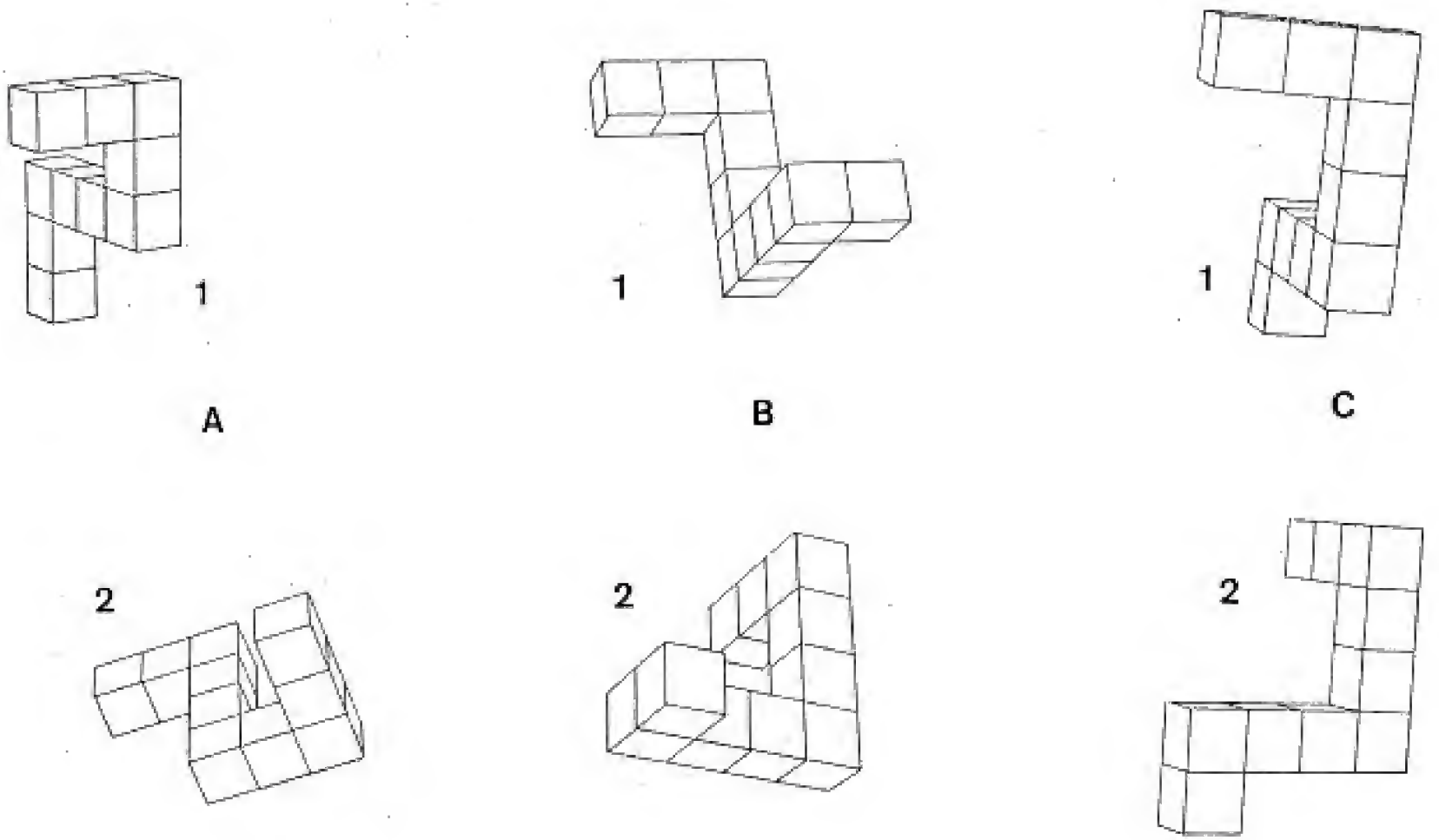


FIGURE 16. Examples of pairs of perspective line drawings presented to the subjects. (A) A "same" pair, which differs by an 80 degree rotation in the picture plane; (B) a "same" pair which differs by an 80 degree rotation in depth; (C) a "different" pair, which cannot be brought into congruence by any rotation. The time taken to decide whether a pair is the "same" varies linearly with the (3-D) angle by which one must be rotated to be brought into correspondence with the other. (reconstructed from figure 1 of Shepard & Metzler, 1971).

rotated as a whole in the image space. The following "mental experiments" will convey the intuitions behind this to the reader. We have confined them to the discussion, because we do not regard such evidence as admissible in the debate about the psychological correctness of the theory.

(a) Imagine a horse. [For most people, it is facing either left or right, which we interpret as the initial space-frame configuration.]

(b) Imagine rotating the horse 90 degrees about its torso.

(c) Where is the neck pointing? [Most people can answer this easily.]

(d) Now imagine a new horse in the starting configuration, only this particular horse has his legs glued onto the top of his back, pointing upwards. His head and neck are in the usual position. Now rotate the horse 180 degrees about his torso. Where are his legs pointing? Where is the neck pointing? [We find that people either "leave the neck behind" in this rotation, and have to reconstruct it afterwards; or they find themselves interrupting the reconstruction of the legs after the rotation because this reconstruction differs from the normal one which they had (by habit) initiated. We think of the actual rotation as having taken place with the Saxis bound to the torso, and the Sspasar either to an "undercarriage" Satom, or to a Satom for the forelegs or the hindlegs, or to a Satom for the neck, or head + neck ("bust").]

(4) The number of possible rotations in our model is small, only four being allowed. [This seems to be true of mental rotations. For example, imagine a normal horse once again, whose tail is about 20 degrees to the vertical. It seems to be straightforward to execute mental rotations about the three principal axes (the horse's up, front, and horizontal); but not about arbitrary directions. For example, try rotating it about its tail One either fails, or has to resort to a special stratagem. If however one imagines the horse standing on a 20 degree slope so that the tail actually falls down the gravitational vertical, it becomes easy to imagine rotating it about the vertical - i.e. about the same axis relative to the horse that was previously so difficult.]

(5) The 3-D model is loosely hierarchical. [In the previous example, one might have thought that one could move the Saxis to the tail, the Sspasar to the torso, and then rotate the horse. This possibility would be excluded if the tail Satom contained no adjunct relation for the torso. Adjunct relations are not symmetric, and this is what in our theory produces the directional property of the hierarchy.]

(6) There is a general agreement between our expectations and the results surveyed by Shepard (1975). Only one of the findings (item 14 page 100) is unexpected. It comes from Cooper & Shepard (1973b condition D), who showed that advance information giving the orientation but not the identity of the object to be presented is not sufficient to enable subjects to prepare for it. One might have expected that subjects could rotate their Sspasar to the appropriate orientation, and leave it there to be bound to a 3-D model when the image was presented. In order to incorporate this finding, we would need to assume (for example) that the Sspasar machinery cannot be run unless bound to a 3-D model (even if only of an arrow), and that whenever the Sspasar is rebound to a new 3-D model, the image-space processor is reset. There are some other grounds for wanting this. The space-frame in the

image-space processor needs more than one direction to define it, and trying to construct a space-frame round a given vector can lead to problems if the 3-D model is not simple. Secondly, in the real world, one rarely sees two objects at the same point in the field of view. Therefore, to change to a new 3-D model almost always requires a change in the direction of gaze. In order to compensate for this in a minimal implementation, the $axis and $spasar would have to be set to axes in the starting frame, in order to carry out the primary rotations that allow for the angle of gaze. These arguments are however weaker than the arguments that support the rest of the theory.

The reader can amuse himself by constructing mental rotation problems, and by devising strategies to answer questions on which he fails the first time. By noticing and exploiting extra relations between the parts of a structure, one can quickly become much more versatile at answering questions about the appearance of an object when rotated in a new way. It appears to us that the mechanisms contained in our theory can account for most of the experiences that one has when imagining such things, and this is partly why we find the theory interesting. But we are perfectly aware that this kind of evidence cannot establish that the theory provides a correct or even an adequate model of this component of our perceptual faculties. What we *do* claim for the theory is that the computational facilities it describes are useful for recognizing and representing the disposition of an object in three-dimensional space.

2: Broader issues

It is no accident that the term "frame" (in the sense of Minsky 1975) has not appeared in this article. We have been careful to use only technical terms ($atom, template) that have a precise meaning in our theory and are supported by a working program. Nevertheless some of our ideas have been influenced (positively or negatively) by Minsky's extraordinarily stimulating (and frustrating) article, and we must attempt to relate our work to the ideas of frame theory.

At the very top level, Minsky must surely be correct when he observed that the "chunks" of reasoning, language, memory and perception ought to be larger and more structured than most theories in artificial intelligence and psychology allowed. Failure to realize this led to absurd attempts to "prove" from predicate-calculus-like "axioms" the "correctness" of strategies for passing between two rooms or circumnavigating an obstacle. Reaction to that line of thought led to the procedural embedding of knowledge, to new languages like PLANNER (Hewitt 1969) and thence to CONNIVER (Sussman & McDermott 1972) - a valuable experiment that has run sufficiently long now for the results to be in, and they are conclusively negative (though with much positive spin-off).

Except at this very general level, it is not clear that the ideas of frame theory are relevant to the specific problems of visual perception. Minsky himself has made no claims that frames are relevant to early visual information processing. The grounds for extending this conclusion to later processing are as follows:

(1) Gestalt phenomena. It is probably incorrect to think of frame theory as being important for perceptual Gestalt phenomena. The Kanizsa triangle, and "sun" illusions like figure 9a

of Marr (1975 a) are not caused by the descending influences of a high-level "frame-like" organization of the percept; they are due to general-purpose intermediate-level grouping processes that act on the primal sketch and together perform much of figure-ground separation (see also Warrington & Taylor 1973 p. 154). Recent work by S. Ullman (personal communication) demonstrates that the same is true for motion vision. Figure-ground separation by relative motion is not caused by extensive top-down matching from a "frame" to the image. It is due almost entirely to local matching processes that operate on the changing primal sketch, and it takes place before any description of the separated figure is computed. Similar remarks hold for percepts due only to binocular disparity information (Julesz 1971), and for the recognition of symmetry in a figure (Marr 1976).

(2) Multiple view representation. It is difficult to argue cogently against this representation, because it is at present underdefined – for example, are all "views" of a man the same in which the same limbs are visible but arranged in different positions? Nevertheless, something of a case against it can be made from Warrington & Taylor's (1973) findings. The side view of a bucket is very different from the top view, and both are reasonably simple. One would expect the multiple view representation to contain them both, and (presumably) to have indexed both of them. If Warrington & Taylor's lesions had randomly damaged the multiple view representation, one would expect some patients to have lost one view, and others, another. But the finding is that all patients are impaired on the same view, and that for this and other objects (e.g. a clarinet), the lost views are precisely those furthest removed from the objects' generalized cylinder representations. Although the multiple view representation is not absolutely incompatible with these findings, strong extra assumptions are needed to incorporate them. On the other hand, they are a natural consequence in our theory of losing the image-space processor.

(3) Frame technology. The major difficulty in criticizing other aspects of frame theory is that they have not yet been made specific enough to be refutable. For example, existing expositions of the theory fail to define what "frames", "terminals", "slots" and "semantic nets" could actually mean beyond the old ideas of property-lists, values and defaults (like Raphael's), and some sort of labelled graph structure – all of which are useful ideas but which are too simple to carry the load attached to their roles in frame theory. We encountered one problem that might be related to the notion of "terminals" on a frame. If the "parts" of one of our horse 3-D models are somehow construed as Minskian terminals on a horse "frame", then one can perhaps make some correspondence between entries in our packet indexes and the terminals of frame theory. The analogy is flawed, because our packets are based on reference evaluation not on matching (a distinction we regard as crucial), and in any case we found that fixed terminals proved too inflexible to be useful. We needed a system in which there were many ways of describing the parts of an animal, and the particular ones chosen (e.g. forelegs, or left-foreleg and right-foreleg, or sometimes both) depended on the circumstances. The real issue, as Fahlman grasped early on, centers on creating this flexibility, which in turn places the reference-window problem and its associated indexing strategies firmly at the center of the debate. Like Woods (1975), we found the idea of a semantic net too vague to be useful; differential diagnosis based on a difference-directed

memory violates the principle of least commitment, and does not appear to be relevant to the type of recognition that we have been studying.

Acknowledgements: D. M. thanks Dr. S. Brenner F. R. S. for important conversations in 1972 about reference and naming; Ken Forbus who helped write MEMNON, an experimental program written over the last 18 months to develop the principles on which our interpreter is based; and Drew McDermott for his perspicuous and penetrating criticism. We emphasize that we could not have developed the theory without the experience of implementing it. This in turn would not have been possible without the extensive and flexible computing facilities that are available at this laboratory. Karen Prendergast prepared the drawings. Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract number N00014-75-C-0643.

References

Agin, G. J. (1972). Representation and description of curved objects. *Stanford A. I. Memo 173.*

Chien, R. T. & Jones, V. C. (1975). Acquisition of moving objects and hand-eye coordination. *International Joint Conference on Artificial Intelligence 1975, vol. 2,* 737-740.

Cooper, L. A. & Shepard, R. N. (1973 a). The time required to prepare for a rotated stimulus. *Memory and Cognition, 1,* 246-250.

Cooper, L. A. & Shepard, R. N. (1973 b). Chronometric studies of the rotation of mental images. In: W. G. Chase (Ed.), *Visual information processing.* New York: Academic Press.

Fahlman, S. E. (1975). A system for representing and using real-world knowledge. *M. I. T. A. I. Lab. Memo 331.*

Hewitt, C. (1969). PLANNER: a language for proving theorems and manipulating models in a robot. *Proceedings of the International Joint Conference on Artificial Intelligence,* pp 295-301. Bedford, Mass: Mitre Corp.

Hollerbach, J. M. (1975). Hierarchical shape description by selection and modification of prototypes. *M. I. T. Master's Thesis, to appear as M. I. T. A. I. Lab. TR-346.*

Julesz, B. (1971). *Foundations of Cyclopean Perception.* Chicago: The University of Chicago Press.

Kosslyn, S. M. (1975). Information representation in visual images. *Cognitive Psychology, 7*, 341-370.

McDermott, D. V. (1975). Very large PLANNER-type data bases. *M. I. T. A. I. Lab. Memo 339.*

Marr, D. (1975 a). Early processing of visual information. *M. I. T. A. I. Lab. Memo 340.*

Marr, D. (1975 b). Less about symbol-mapping. *ACM Sigart Newsletter, 55*, (to appear).

Marr, D. (1976). Intermediate processing of visual information. (In preparation)

Metzler, J. & Shepard, R. N. (1974). Transformational studies of the internal representation of three-dimensional studies. In: *Theories of cognitive psychology: The Loyola Symposium*, Ed. R. Solso. Hillsdale, N. J.: Lawrence Erlbaum Assoc.

Minsky, M. (1975). A framework for representing knowledge. In: *The psychology of computer vision*, Ed. P. H. Winston, pp 211-277. New York: McGraw-Hill.

Minsky, M. & Papert, S. (1972). Artificial intelligence progress report. *M. I. T. A. I. Lab. Memo 252.*

Nevatia, R. (1974). Structured descriptions of complex curved objects for recognition and visual memory. *Stanford A. I. Memo 250.*

Raphael, B. (1968). SIR: semantic information retrieval. In: *Semantic information processing*, Ed. M. Minsky, pp 33-145.

Rubin, A. D. (1975). Hypothesis formation and evaluation in medical diagnosis. *M. I. T. Technical Report 316.*

Shepard, R. N. (1975). Form, formation, and transformation of internal representations. In: *Information processing and cognition: The Loyola Symposium*, Ed. R. Solso, pp 87-122. Hillsdale, N. J.: Lawrence Erlbaum Assoc.

Shepard, R. N. & Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science, 171, 701-703.*

Speckert, G. (1975). Visual tracking of real world objects. *M. I. T. A. I. Lab. Working Paper 106.*

Sussman, G. J. & McDermott, D. V. (1972). From PLANNER to CONNIVER - a genetic approach.

Proceedings of the Fall Joint Computer Conference, 41, 1171-1179.

Warrington, E. K. & Taylor, A. M. (1973). The contribution of the right parietal lobe to object recognition. *Cortex*, 9, 152-164.
Also remarks made by E. K. W. in a lecture given on Oct. 26th 1973 at the M. I. T. Psychology Department.

Woods, W. A. (1975). What's in a link: foundations for semantic networks. In: *Representation and understanding*, Eds. Bobrow, D. G. & Collins, A., pp 35-82. New York: Academic Press.